Engineering
Simplicity

Junos® OS

Adaptive Services Interfaces User Guide for Routing Devices

Published
2024-02-21

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

Junos® OS Adaptive Services Interfaces User Guide for Routing Devices

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at https://support.juniper.net/support/eula/. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

About This Guide | xxi

1 Overview

Services Overview | 2

Adaptive Services and Multiservices Interfaces Overview | 2

Packet Flow Through the Adaptive Services or Multiservices PIC | 4

Services Configuration Overview | 7

Service Sets | 7

- Understanding Service Sets | 8
- Configuring Service Sets to be Applied to Services Interfaces | 10
- Configuring Service Set Limitations | 15
- Example: Configuring Service Sets | 17
- Configuring Service Interface Pools | 17
- Enabling Services PICs to Accept Multicast Traffic | 17
- Applying Filters and Services to Interfaces | 18
- Examples: Configuring Services Interfaces | 21
- Configuring the Address and Domain for Services Interfaces | 23
- Configuring System Logging for Service Sets | 24
- Configuring Service Rules | 27

TCP Fast Open | 28

- Exchanging Data More Efficiently Using TCP Fast Open | 28
- Configuring TFO | 30
  - Three Modes for TFO | 30
  - Using NAT and TFO | 33
- Configuring Fragmentation Control for MS-DPC and MS-PIC Service Interfaces | 34
- Tracing Services PIC Operations | 35

Service Filters | 39

- Service Filters in ACX Series | 39
- Guidelines for Applying Service Filters | 40
- Service Filter Match Conditions for IPv4 Traffic | 42

Service Filter Actions | 43

Applying Filters and Services to Interfaces | 44

Configuring Queuing and Scheduling on Inline Services Interface | 47

Configuring the Address and Domain for Services Interfaces | 49

Enabling Session Offloading for Multiservices DPCs | 50

2 Network Address Translation

NAT Overview | 53

Network Address Translation Overview | 53

Junos Address Aware Network Addressing Overview | 53

Sample IPv6 Transition Scenarios | 62

Junos OS Carrier-Grade NAT Implementation Overview | 64

Carrier-Grade NAT Feature Comparison for Junos Address Aware by Type of Interface Card | 64

ALGs Available for Junos OS Address Aware NAT | 71

ALGs Available by Default for Junos OS Address Aware NAT on ACX500 Router | 77

NAT Configuration Overview | 92

Network Address Translation Configuration Overview | 92

Configuring Source and Destination Addresses Network Address Translation Overview | 92

Configuring Pools of Addresses and Ports for Network Address Translation Overview | 94

Network Address Translation Rules Overview | 97

Protecting CGN Devices Against Denial of Service (DOS) Attacks | 106

Carrier-Grade NAT Implementation: Best Practices | 106

Network Address Translation Overview on ACX Series | 120

Network Address Translation Overview on ACX Series | 120

Network Address Port Translation Overview | 121

Network Address Translation Address Overload in ACX Series | 122

Network Address Translation Constraints on ACX | 123

Enabling Inline Services Interface on ACX Series | 124

Stateful NAT64 | 126

Stateful NAT64 | 126

Configuring Stateful NAT64 | 126

Static Source NAT | 131

Static Source NAT | 131

Configuring Static Source Translation in IPv4 Networks | 131

Configuring the NAT Pool and Rule | 132

Configuring the Service Set for NAT | 135

Configuring Trace Options | 137

Sample Configuration - Static Source NAT Using a Static Pool With An Address Prefix And An Address Range | 139

Sample Configuration - Static Source Nat for One-To-One Mapping Between a Private Subnet and a Public Subnet | 139

Configuring Static Source Translation in IPv6 Networks | 141

Configuring the NAT Pool and Rule | 141

Configuring the Service Set for NAT | 143

Configuring Trace Options | 145

Example: Configuring Basic NAT44 | 147

Requirements | 147

Overview | 147

Configuring Basic NAT44 | 147

Example: Configuring NAT for Multicast Traffic | 150

Static Destination NAT | 156

Static Destination NAT | 156

Configuring Static Destination Address Translation in IPv4 Networks | 156

Network Address Port Translation | 163

Network Address Port Translation | 163

Configuring Address Pools for Network Address Port Translation (NAPT) Overview | 163

Round-Robin Allocation for NAPT | 164

Sequential Allocation for NAPT | 165

Preserve Parity and Preserve Range for NAPT | 166

Address Pooling and Endpoint Independent Mapping for NAPT | 166

Secured Port Block Allocation for NAPT | 168

Comparison of NAPT Implementation Methods | 169

Configuring NAPT in IPv4 Networks | 170

Configuring NAPT in IPv6 Networks | 176

Example: Configuring NAT with Port Translation | 179

Requirements | 180

Overview | 180

Configuring NAT with Port Translation | 180

Example: NAPT Configuration on the MS-MPC With an Interface Service Set | 183

Requirements | 183

Overview | 183

Configuration | 183

Deterministic NAT | 191

Deterministic NAT | 191

Deterministic NAPT Overview | 191

Configuring Deterministic NAPT | 197

Configuring the NAT Pool for Deterministic NAPT | 197

Configuring the NAT Rule for Deterministic NAPT | 199

Configuring the Service Set for Deterministic NAT | 200

NAT Protocol Translation | 202

NAT Protocol Translation | 202

Configuring NAT-PT | 202

Configuring the DNS ALG Application | 203

Configuring the NAT Pool and NAT Rule | 203

Configuring the Service Set for NAT | 208

Configuring Trace Options | 210

Example: Configuring NAT-PT | 212

Requirements | 213

Overview and Topology | 213

Configuration of NAT-PT with DNS ALGs | 215

Example: Configuring the DNS ALG Application on MX-SPC3 service card | 233

IPv4 Connectivity Across IPv6-Only Network Using 464XLAT | 238

IPv4 Connectivity Across IPv6-Only Network Using 464XLAT | 238

464XLAT Overview | 238

Configuring 464XLAT Provider-Side Translater for IPv4 Connectivity Across IPv6-Only Network | 240

Port Control Protocol | 243

Port Control Protocol | 243

Port Control Protocol Overview | 243

Configuring Port Control Protocol | 246

Configuring PCP Server Options | 247

Configuring a PCP Rule | 249

Configuring a NAT Rule | 250

Configuring a Service Set to Apply PCP | 251

SYSLOG Message Configuration | 252

Monitoring Port Control Protocol Operations | 252

Example: Configuring Port Control Protocol with NAPT44 | 254

Requirements | 254

Overview | 255

PCP Configuration | 255

Secured Port Block Allocation | 264

Secured Port Block Allocation | 264

Secured Port Block Allocation for NAPT44 and NAT64 Overview | 264

Guidelines for Configuring Secured Port Block Allocation | 265

Configuring Secured Port Block Allocation | 267

Secured Port Block Allocation Interim Logging | 271

Interim Logging for Secured Port Block Allocation | 271

Guidelines for Configuring Interim Logging for Secured Port Block Allocation | 272

Port Forwarding | 276

Port Forwarding | 276

Port Forwarding Overview | 276

Configuring Port Forwarding for Static Destination Address Translation | 277

Configuring Port Forwarding Without Destination Address Translation | 281

Example: Configuring Port Forwarding with Twice NAT | 284

Dynamic Address-Only Source Translation | 287

Dynamic Address-Only Source Translation | 287

Configuring Dynamic Address-Only Source Translation in IPv4 Networks | 287

Example: Dynamic Source NAT as a Next-Hop Service | 294

Example: Assigning Addresses from a Dynamic Pool for Static Use | 296

Inline NAT | 298

Inline NAT | 298

Inline Network Address Translation Overview | 298

Example: Configuring Inline Network Address Translation—Interface-Based Method | 300

Requirements | 300

Overview and Topology | 301

Configuration | 302

Verification | 307

Example: Configuring Inline Network Address Translation—Route-Based Method | 309

Requirements | 310

Overview and Topology | 310

Configuration | 311

Verification | 317

Stateless Source Network Prefix Translation for IPv6 | 321

Stateless Source Network Prefix Translation for IPv6 | 321

Stateless Source Network Prefix Translation for IPv6 Overview | 321

Interoperation of Functionalities with Network Prefix Translation for IPv6 | 323

Guidelines for Configuring Stateless Source Network Prefix Translation | 326

Working of NPTv6 with Interface-Style and Next Hop-Style Service Sets | 327

Example: Achieving Address Independence By Configuring Stateless Network Prefix Translation in IPv6 Networks by Using Interface-Style Service Sets | 328

Requirements | 329

Overview and Topology for Stateless Network Prefix Translation in IPv6 Networks Using Interface-Style Service Sets | 329

Configuration | 329

Verification | 334

Example: Achieving Address Independence By Configuring Stateless Network Prefix Translation in IPv6 Networks by Using Next-Hop -Style Service Sets | 336

Requirements | 337

Overview and Topology of Stateless Network Prefix Translation in IPv6 Networks Using Next-Hop Style Service Sets | 337

Configuration | 337

Verification | 344

Monitoring NAT | 347

Monitoring NAT | 347

Configuring NAT Session Logs | 347

Monitoring NAT Pool Usage | 349

Using the Enterprise-Specific Utility MIB | 350

Using the Enterprise-Specific Utility MIB | 350

Populating the Enterprise-Specific Utility MIB with Information | 351

Stopping the SLAX Script with the CLI | 358

Clearing the Utility MIB | 358

Recovering from an Abnormal SLAX Script Exit or a SLAX Script Exit with the CLI | 359

Packet Translation and GRE Tunneling | 360

Packet Translation and GRE Tunneling | 360

Packet Translation and GRE Tunneling-Overview | 360

Encapsulation Process (Edge Router-to-PaaS Server) | 361

De-encapsulation Process (PaaS Server to Edge Router) | 365

3 **Transitioning to IPv6 Using MAP-E and MAP-T**

Transitioning to IPv6 Using MAP-E and MAP-T | 371

Mapping of Address and Port with Encapsulation (MAP-E) | 371

Understanding Mapping of Address and Port with Encapsulation (MAP-E) | 371

Configuring Mapping of Address and Port with Encapsulation (MAP-E) | 375

Equal Cost Multiple Path (ECMP) support for Mapping of Address and Port with Encapsulation (MAP-E) | 380

Equal Cost Multiple Path (ECMP) support for Mapping of Address and Port with Encapsulation (MAP-E) | 381

Disabling auto-routes to support ECMP with Mapping of Address and Port with Encapsulation (MAP-E) | 381

Mapping of Address and Port with Translation (MAP-T) | 385

Understanding Mapping of Address and Port with Translation (MAP-T) | 385

Configuring Mapping of Address and Port using Translation (MAP-T) | 387

4 **Transition to IPv6 With Softwires**

Transition to IPv6 With 6to4 Softwires | 394

Softwires Configuration Overview | 394

Tunneling Services for IPv4-to-IPv6 Transition Overview | 394

Configuring Softwire Rules | 399

Configuring Service Sets for Softwire | 400

6to4 Softwires | 402

Configuring a 6to4 Provider-Managed Tunnel | 402

Transition to IPv6 With DS-Lite Softwires | 407

DS-Lite Softwires | 407

Configuring a DS-Lite Softwire Concentrator | 407

Configuring IPv6 Multicast Interfaces | 408

Example: Basic DS-Lite Configuration | 409

Requirements | 409

Configuration Overview and Topology | 410

Configuration | 411

Example: Configuring DS-Lite and 6rd in the Same Service Set | 418

Requirements | 418

Overview | 418

Configuration | 418

DS-Lite Subnet Limitation | 427

DS-Lite Per Subnet Limitation Overview | 427

Configuring DS-Lite Per Subnet Session Limitation to Prevent Denial of Service Attacks | 429

Transition to IPv6 With 6rd Softwires | 432

Configuring a 6rd Softwire | 432

Configuring a 6rd Softwire Concentrator | 432

Configuring Stateful Firewall Rules for 6rd Softwire | 433

Example: Basic 6rd Configuration | 435

Requirements | 435

Overview | 435

Configuration | 435

High Availability and Load Balancing for 6rd Softwires | 442

Load Balancing a 6rd Domain Across Multiple Services PICs | 442

Example: Load Balancing a 6rd Domain Across Multiple Services PICs | 442

Configuring High Availability for 6rd Using 6rd Anycast | 450

Transition to IPv6 With Inline Softwires | 451

Inline 6rd and 6to4 Softwires | 451

Inline 6rd and 6to4 Configuration Guidelines | 451

Configuring Inline 6rd | 452

Configuring the Bandwidth for Inline Services | 453

Configuring the Interfaces | 453

Configuring the Softwire Concentrator and Rule | 455

Configuring the Service Set | 457

Configuring the Routing Instance | 457

Examples: 6rd and 6to4 Configurations | 458

Monitoring and Troubleshooting Softwires | 469

Monitoring and Troubleshooting Softwires | 469

Ping and Traceroute for DS-Lite | 469

Monitoring Softwire Statistics | 470

Monitoring CGN, Stateful Firewall, and Softwire Flows | 472

5 ALGs

ALGs | 475

ALG Overview | 475

ALG Descriptions | 475

ICMP, Ping, and Traceroute ALGs for MS-MICs and MS-MPCs | 506

ALG Applications | 507

Configuring Application Properties | 508

Configuring Application Sets | 532

Examples: Configuring Application Protocols | 532

Verifying the Output of ALG Sessions | 533

6 Access Security

Stateful Firewalls | 546

Stateful Firewalls | 546

Junos Network Secure Overview | 546

Configuring Stateful Firewall Rules | 550

Configuring Stateful Firewall Rule Sets | 556

Examples: Configuring Stateful Firewall Rules | 557

Example: BOOTP and Broadcast Addresses | 561

Example: Configuring Layer 3 Services and the Services SDK on Two PICs | 562

Example: Virtual Routing and Forwarding (VRF) and Service Configuration | 583

Monitoring Stateful Firewalls | 586

Monitoring Stateful Firewall Conversations | 586

Monitoring Global Stateful Firewall Statistics | 587

IDS on MS-DPC | 588

IDS on MS-DPC | 588

Understanding SYN Cookie Protection on an MS-DPC | 588

Configuring IDS Rules on an MS-DPC | 590

Configuring IDS Rule Sets on an MS-DPC | 600

Examples: Configuring IDS Rules on an MS-DPC | 601

Network Attack Protection on MS-MPC and MS-MIC | 605

Network Attack Protection on MS-MPC and MS-MIC | 605

Understanding IDS on an MS-MPC | 605

Configuring Protection Against Network Attacks on an MS-MPC | 610

Configuring Protection Against Network Probing, Network Flooding, and Suspicious Pattern Attacks | 610

Configuring Protection Against Header Anomaly Attacks | 620

Configuring Logging of Network Attack Protection Packet Drops on an MS-MPC | 621

7 **IPsec Tunnels**

IPsec Overview | 623

IPsec Overview | 623

Understanding Junos VPN Site Secure | 623

Authentication Algorithms | 627

Encryption Algorithms | 628

IPsec Protocols | 629

IPsec Multipath Forwarding with UDP Encapsulation | 632

Supported IPsec and IKE Standards | 634

IPSec Terms and Acronyms | 636

Triple Data Encryption Standard (3DES) | 637

Adaptive Services PIC | 637

Advanced Encryption Standard (AES) | 637

authentication header (AH) | 637

certificate authority (CA) | 637

certificate revocation list (CRL) | 637

cipher block chaining (CBC) | 637

Data Encryption Standard (DES) | 638

digital certificate | 638

ES PIC | 638

Encapsulating Security Payload (ESP) | 638

Hashed Message Authentication Code (HMAC) | 638

Internet Key Exchange (IKE) | 638

Message Digest 5 (MD5) | 638

Perfect Forward Secrecy (PFS) | 638

public key infrastructure (PKI) | 639

registration authority (RA) | 639

Routing Engine | 639

security association (SA) | 639

Security Association Database (SADB) | 639

Secure Hash Algorithm 1 (SHA-1) | 639

Secure Hash Algorithm 2 (SHA-2) | 639

Security Policy Database (SPD) | 639

Security Parameter Index (SPI) | 640

Simple Certificate Enrollment Protocol (SCEP) | 640

IPsec for ACX Series Overview | 640

IPsec Tunnels With Static Endpoints | 643

Minimum Security Association Configurations | 643

Configuring Security Associations | 645

Configuring Manual Security Associations | 646

Configuring the Direction for IPsec Processing | 647

Configuring the Protocol for a Manual IPsec SA | 648

Configuring the Security Parameter Index | 649

Configuring the Auxiliary Security Parameter Index | 649

Configuring Authentication for a Manual IPsec SA | 650

Configuring Encryption for a Manual IPsec SA | 650

Configuring Dynamic Security Associations | 651

Clearing Security Associations | 652

Manual Security Associations | 653

Example: Configuring Manual SAs | 653

Requirements | 654

Overview and Topology | 654

Configuration | 655

Verification | 671

Dynamic Security Associations | 675

Configuring IKE Proposals | 675

Configuring the Authentication Algorithm for an IKE Proposal | 676

Configuring the Authentication Method for an IKE Proposal | 677

Configuring the Diffie-Hellman Group for an IKE Proposal | 678

Configuring the Encryption Algorithm for an IKE Proposal | 679

Configuring the Lifetime for an IKE SA | 679

Example: Configuring an IKE Proposal | 680

Configuring IKE Policies | 681

Configuring the IKE Phase | 682

Configuring the Mode for an IKE Policy | 683

Configuring the Proposals in an IKE Policy | 683

Configuring the Preshared Key for an IKE Policy | 683

Configuring the Local Certificate for an IKE Policy | 684

Configuring the Description for an IKE Policy | 685

Configuring Local and Remote IDs for IKE Phase 1 Negotiation | 685

Enabling Invalid SPI Recovery | 687

Example: Configuring an IKE Policy | 687

Configuring IPsec Proposals | 688

Configuring the Authentication Algorithm for an IPsec Proposal | 689

Configuring the Description for an IPsec Proposal | 691

Configuring the Encryption Algorithm for an IPsec Proposal | 691

Configuring the Lifetime for an IPsec SA | 692

Configuring the Protocol for a Dynamic SA | 693

Configuring IPsec Policies | 694

Configuring the Description for an IPsec Policy | 695

Configuring Perfect Forward Secrecy | 695

Configuring the Proposals in an IPsec Policy | 696

IPsec Policy for Dynamic Endpoints | 696

Example: Configuring an IPsec Policy | 697

IPsec Rules and Rulesets | 699

Example: Configuring IKE Dynamic SAs | 699

Requirements | 699

Overview and Topology | 700

Configuration | 701

Verification | 717

Configuring IPsec Rules | 723

Configuring Match Direction for IPsec Rules | 725

Configuring Match Conditions in IPsec Rules | 726

Configuring Actions in IPsec Rules | 728

Configuring IPsec Rule Sets | 732

Service Sets for Static Endpoint IPsec Tunnels | 733

Service Sets | 733

Configuring IPsec Service Sets | 734

Requesting for and Installing a Digital Certificates on Your Router | 742

Requesting a Digital Certificate—Manual Process | 743

Example: IKE Dynamic SA Configuration with Digital Certificates | 745

Requirements | 746

Overview | 746

Configuration | 747

Verification | 765

Configuring Junos VPN Site Secure or IPSec VPN | 777

Example: Configuring Junos VPN Site Secure on MS-MIC and MS-MPC | 778

Requirements | 778

Overview | 779

Configuration | 780

Verification | 791

Example: Configuring Statically Assigned IPsec Tunnels over a VRF Instance | 793

Requirements | 794

Overview | 794

Configuration | 794

Multitask Example: Configuring IPsec Services | 801

Configuring the IKE Proposal | 801

Configuring the IKE Policy (and Referencing the IKE Proposal) | 803

Configuring the IPsec Proposal | 804

Configuring the IPsec Policy (and Referencing the IPsec Proposal) | 805

Configuring the IPsec Rule (and Referencing the IKE and IPsec Policies) | 806

Configuring IPsec Trace Options | 807

Configuring the Access Profile (and Referencing the IKE and IPsec Policies) | 808

Configuring the Service Set (and Referencing the IKE Profile and the IPsec Rule) | 810

Disabling NAT-T on MX Series Routers for Handling NAT with IPsec-Protected Packets | 811

Tracing Junos VPN Site Secure Operations | 812

Disabling IPsec Tunnel Endpoint in Traceroute | 814

Tracing IPsec PKI Operations | 814

IPsec Tunnels With Dynamic Endpoints | 817

IPsec Tunnels With Dynamic Endpoints | 817

Configuring Dynamic Endpoints for IPsec Tunnels | 817

Authentication Process | 818

Implicit Dynamic Rules | 818

Reverse Route Insertion | 819

Configuring an IKE Access Profile | 820

Referencing the IKE Access Profile in a Service Set | 821

Configuring the Interface Identifier | 822

Default IKE and IPsec Proposals | 822

Distributing Endpoint IPsec Tunnels Among Services Interfaces | 823

Example: Configuring Dynamically Assigned Policy Based Tunnels | 825

Requirements | 825

Overview and Topology | 825

Configuration | 826

Verification | 832

8

CoS on Services Cards

CoS on Services Cards | 835

Class of Service on Services Interfaces | 835

Class of Service Overview | 835

Restrictions and Cautions for CoS Configuration on Services Interfaces | 836

Configuring CoS Rules | 837

Configuring CoS Rule Sets | 843

Examples: Configuring CoS on Services Interfaces | 844

Class of Service on Link Services Interfaces | 847

Class of Service on Link Services Interfaces | 847

Link Services Configuration for Junos Interfaces | 847

Configuring CoS Scheduling Queues on Logical LSQ Interfaces | 849

Configuring CoS Fragmentation by Forwarding Class on LSQ Interfaces | 853

Configuring Link Services and CoS on Services PICs | 856

Oversubscribing Interface Bandwidth on LSQ Interfaces | 860

Configuring Guaranteed Minimum Rate on LSQ Interfaces | 866

9 Inter-Chassis Redundancy for NAT and Stateful Firewall Flows

Configuring Inter-Chassis MS-MPC and MS-MIC for NAT and Stateful Firewall (Release 16.1 and later) | 873

Inter-Chassis Stateful Synchronization for Long Lived NAT and Stateful Firewall Flows (MS-MPC, MS-MIC) (Release 16.1 and later) | **873**

Configuring Inter-chassis MS-MPC and MS-MIC Redundancy for NAT and Stateful Firewall Overview (Release 16.1 and later) | **874**

Inter-Chassis Stateful Synchronization for Long Lived NAT and Stateful Firewall Flows (MS-MPC, MS-MIC) Overview (Release 16.1 and later) | **874**

Configuring Inter-Chassis Stateful Synchronization for Long Lived NAT and Stateful Firewall Flows (MS-MPC, MS-MIC) (Release 16.1 and later) | **876**

Example: Inter-Chassis Stateful Synchronization for Long-Lived NAT and Stateful Firewall Flows (MS-MIC, MS-MPC) (Release 16.1 and later) | **878**

Requirements | **878**

Overview | **878**

Configuration | **879**

Service Redundancy Daemon | **890**

Service Redundancy Daemon Overview | **891**

Configuring the Service Redundancy Daemon | **893**

Configuring Redundancy Events | **895**

Configuring Redundancy Policies | **896**

Configuring Redundancy Set and Group | **898**

Configuring Routing Policies Supporting Redundancy | **900**

Configuring Service Sets | **901**

Using Service Redundancy Daemon Scripts to View and Change the Status of a Gateway | **901**

Configuring Inter-Chassis Stateful Synchronization for NAT and Stateful Firewall (Release 15.1 and earlier) | 903

Inter-Chassis High Availability for MS-MIC and MS-MPC (Release 15.1 and earlier) | **903**

Inter-Chassis High Availability for Stateful Firewall and NAPT44 Overview (MS-MIC, MS-MPC) | **904**

Configuring Inter-Chassis High Availability for Stateful Firewall and NAPT44 (MS-MPC, MS-MIC) | **905**

Example: Inter-Chassis Stateful High Availability for NAT and Stateful Firewall (MS-MIC, MS-MPC) | **906**

Requirements | **906**

Overview | **907**

Configuration | **907**

10 Multilinks

Link Services Interface Redundancy | 921

Link Services Interface Redundancy | 921

Layer 2 Service Package Capabilities and Interfaces | 921

Configuring LSQ Interface Redundancy Across Multiple Routers Using SONET APS | 923

Configuring LSQ Interface Redundancy in a Single Router Using SONET APS | 926

Configuring LSQ Interface Redundancy in a Single Router Using Virtual Interfaces | 927

Link Bundling | 939

Inline Multlink Services | 939

Inline MLPPP for WAN Interfaces Overview | 939

Reserving Bundle Bandwidth for Link-Layer Overhead on LSQ Interfaces | 942

Enabling Inline LSQ Services | 943

Configuring LSQ Interfaces as NxT1 or NxE1 Bundles Using MLPPP | 945

Configuring LSQ Interfaces as NxT1 or NxE1 Bundles Using FRF.16 | 952

Configuring LSQ Interfaces as NxT1 or NxE1 Bundles Using FRF.15 | 959

Configuring LSQ Interfaces for Single Fractional T1 or E1 Interfaces Using MLPPP and LFI | 960

Configuring LSQ Interfaces for Single Fractional T1 or E1 Interfaces Using FRF.12 | 966

Configuring LSQ Interfaces for T3 Links Configured for Compressed RTP over MLPPP | 976

Configuring LSQ Interfaces as T3 or OC3 Bundles Using FRF.12 | 977

Configuring LSQ Interfaces for ATM2 IQ Interfaces Using MLPPP | 980

11 Traffic Load Balancer

Traffic Load Balancer | 984

Traffic Load Balancer | 984

Traffic Load Balancer Overview | 984

Configuring TLB | 993

Loading the TLB Service Package | 994

Configuring a TLB Instance Name | 994

Configuring Interface and Routing Information | 995

Configuring Servers | 998

Configuring Network Monitoring Profiles | 998

Configuring Server Groups | 1000

Configuring Virtual Services | 1002

Configuring Tracing for the Health Check Monitoring Function | 1005

12

Services Card Redundancy

Services Card Redundancy for MS-MPC and MS-MIC | 1010

Load Balancing and High Availability With Aggregated Multiservices Interfaces on MS-MPC and MS-MIC | 1010

Understanding Aggregated Multiservices Interfaces | 1010

Configuring Aggregated Multiservices Interfaces | 1017

Configuring Load Balancing on AMS Infrastructure | 1020

Configuring Warm Standby for Services Interfaces | 1024

Example: Configuring an Aggregated Multiservices Interface (AMS) | 1025

Hardware and Software Requirements | 1025

Overview | 1025

Configuration | 1026

Verification | 1031

Example: Configuring Next-Hop Style Services on an Aggregated Multiservices Interface | 1032

Hardware and Software Requirements | 1037

Overview | 1037

Example: Configuring Static Source Translation on AMS Infrastructure | 1037

Services Card Redundancy for Multiservices PIC | 1042

Configuring AS or Multiservices PIC Redundancy | 1042

13

Voice Services

Voice Services | 1046

Voice Services | 1046

Voice Services Overview | 1046

Configuring Services Interfaces for Voice Services | 1047

Configuring Encapsulation for Voice Services | 1050

Configuring Network Interfaces for Voice Services | 1051

Examples: Configuring Voice Services | 1052

14

Layer 2 PPP Tunnels

Layer 2 Tunneling of PPP Packets | 1058

Layer 2 Tunneling of PPP Packets | 1058

Layer 2 Tunneling Protocol Overview | 1058

L2TP Services Configuration Overview | 1059

L2TP Minimum Configuration | 1060

Configuring L2TP Tunnel Groups | 1063

Configuring the Identifier for Logical Interfaces that Provide L2TP Services | 1068

AS PIC Redundancy for L2TP Services | 1070

Examples: Configuring L2TP Services | 1071

Tracing L2TP Operations | 1075

15 URL Filtering

URL Filtering | 1079

URL Filtering | 1079

URL Filtering Overview | 1079

Configuring URL Filtering | 1085

DNS Request Filtering for Disallowed Website Domains | 1090

Overview of DNS Request Filtering | 1090

How to Configure DNS Request Filtering | 1092

Multitenant Support for DNS Filtering | 1100

Configuring Multi-tenant Support for DNS Filtering | 1101

Example: Configuring Multitenant Support for DNS Filtering | 1106

Integration of Juniper ATP Cloud and Web Filtering on MX Series Routers | 1111

Overview | 1112

Configuring the Web Filter Profile for Sampling | 1116

GeoIP Filtering | 1121

Global Allowlist and Global Blocklist | 1123

16 Configuration Statements and Operational Commands

unidirectional-session-refreshing | 1127

Junos CLI Reference Overview | 1128

About This Guide

Use this guide to configure and monitor the following services:

- Network Address Translation (NAT)
- Stateful firewalls
- URL filtering
- Intrusion detection service (IDS)
- IP Security (IPsec)
- Application Layer Gateways (ALGs)
- Class of service (CoS) for packets transiting service cards
- Voice services
- Load balancing of server traffic

RELATED DOCUMENTATION

Day One: CGNAT Up and Running on the MX Series

1

PART

Overview

Services Overview | 2

Services Configuration Overview | 7

CHAPTER 1

Services Overview

IN THIS CHAPTER

- Adaptive Services and Multiservices Interfaces Overview | 2
- Packet Flow Through the Adaptive Services or Multiservices PIC | 4

Adaptive Services and Multiservices Interfaces Overview

MultiServices PICs and MultiServices Dense Port Concentrators (MS-DPCs) provide *adaptive services interfaces*, which allow you to coordinate multiple services on a single PIC by configuring a set of services and applications. MultiServices PICs and MS-DPCs offer a special range of services you configure in one or more service sets.

The MultiServices PIC is available in three versions, the MultiServices 100, the MultiServices 400, and the MultiServices 500, which differ in memory size and performance. All versions offer enhanced performance in comparison with AS PICs. MultiServices PICs are supported on M Series and T Series routers except M20 routers.

The MultiServices DPC is available for MX Series routers; it includes a subset of the functionality supported on the MultiServices PIC. Currently the MultiServices DPC supports the following Layer 3 services: stateful firewall, NAT, IDS, IPsec, active flow monitoring, RPM, and generic routing encapsulation (GRE) tunnels (including GRE key and fragmentation); it also supports *graceful Routing Engine switchover* (GRES) and Dynamic Applicaton Awareness for Junos OS. For more information about supported packages, see *Enabling Service Packages*.

It is also possible to group several Multiservices PICs into an aggregated Multiservices (AMS) system. An AMS configuration eliminates the need for separate routers within a system. The primary benefit of having an AMS configuration is the ability to support load balancing of traffic across multiple services PICs. Starting with Junos OS 11.4, all MX Series routers will support high availability (HA) and Network Address Translation (NAT) on AMS infrastructure. See "Configuring Load Balancing on AMS Infrastructure" on page 1020 for more information.

NOTE: The MultiServices PICs are polling based and not interrupt based; as a result, a high value in the `show chassis pic` “Interrupt load average” field may not mean that the PIC has reached its maximum limit of processing.

The following services are configured within a service set and are available only on adaptive services interfaces:

- Stateful firewall—A type of *firewall filter* that considers state information derived from previous communications and other applications when evaluating traffic.
- Network Address Translation (NAT)—A security procedure for concealing host addresses on a private network behind a pool of public addresses.
- Intrusion detection service (IDS)—A set of tools for detecting, redirecting, and preventing certain kinds of network attack and intrusion.
- IP Security (IPsec)—A set of tools for configuring manual or dynamic security associations (SAs) for encryption of data traffic.
- *Class of service* (CoS)—A subset of CoS functionality for services interfaces, limited to DiffServ code point (DSCP) marking and forwarding-class assignment. CoS BA classification is not supported on services interfaces.

The configuration for these services comprises a series of rules that you can arrange in order of precedence as a *rule set*. Each rule follows the structure of a firewall filter, with a `from` statement containing input or match conditions and a `then` statement containing actions to be taken if the match conditions are met.

The following services are also configured on the MultiServices PICs and MS-DPCs, but do not use the rule set definition:

- Layer 2 Tunneling Protocol (L2TP)—A tool for setting up secure tunnels using Point-to-Point Protocol (PPP) encapsulation across Layer 2 networks.
- Link Services Intelligent Queuing (LSQ)—Interfaces that support Junos OS class-of-service (CoS) components, link fragmentation and interleaving (LFI) (FRF.12), Multilink Frame Relay (MLFR) user-to-network interface (UNI) network-to-network interface (NNI) (FRF.16), and Multilink PPP (MLPPP).
- Voice services—A feature that uses the Compressed Real-Time Transport Protocol (CRTP) to enable voice over IP traffic to use low-speed links more effectively.

In addition, Junos OS includes the following tools for configuring services:

- Application protocols definition—Allows you to configure properties of application protocols that are subject to processing by router services, and group the application definitions into application sets.

- Service-set definition—Allows you to configure combinations of directional rules and default settings that control the behavior of each service in the service set.

> **NOTE**: Logging of adaptive services interfaces messages to an external server by means of the **fxp0** port is not supported on M Series routers. The architecture does not support system logging traffic out of a management interface. Instead, access to an external server is supported on a Packet Forwarding Engine interface.

RELATED DOCUMENTATION

Services PICs-Overview

Enabling Service Packages

Services Configuration Procedure

Supported Platforms

Packet Flow Through the Adaptive Services or Multiservices PIC

You can optionally configure service sets to be applied at one of the following three points while the packets transit the router:

- An interface service set applied at the inbound interface.
- A next-hop service set applied at the forwarding table.
- An interface service set applied at the outbound interface.

The packet flow is as follows, graphically displayed in Figure 1 on page 5 . (You can configure a service set as either an interface service set or a next-hop service set.)

1. Packets enter the router on the inbound interface.
2. A policer, filter, service filter, service set, postservice filter, and input forwarding-table filter are applied sequentially to the traffic; these are all optional items in the configuration. If an interface service set is applied, the packets are forwarded to the AS or MultiServices PIC for services processing and then sent back to the Packet Forwarding Engine; if a service filter is also applied, only packets matching the service filter are sent to the PIC. The optional postservice filter is applied and postprocessing takes place.

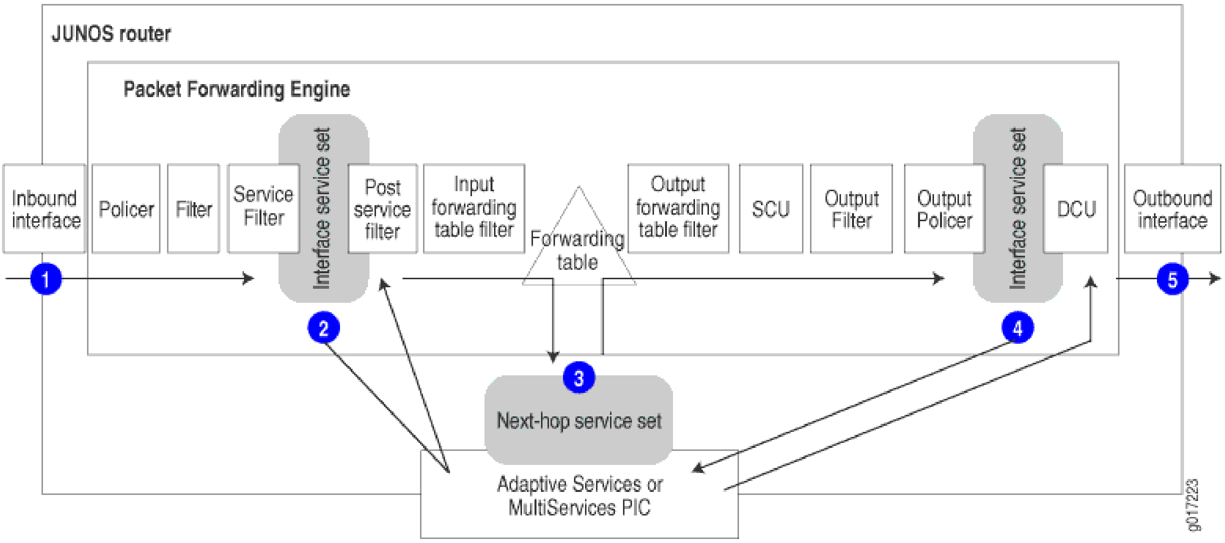
3. A next-hop service set can be applied to the VPN routing and forwarding (VRF) table or to `inet.0`. If it is applied, packets are sent to the PIC for services processing and sent back to the Packet Forwarding Engine.

> **NOTE**: For NAT, the next-hop service set can only be applied to the VRF table. For all other services, the next-hop service set can be applied to either the VRF table or to `inet.0`.

4. On the output interface, an output filter, output policer, and interface service set can be applied sequentially to the traffic if you have configured any of these items. If an interface service set is applied, the traffic is forwarded to the PIC for processing and sent back to the Packet Forwarding Engine, which then forwards the traffic.

5. Packets exit the router.

Figure 1: Packet Flow Through the Adaptive Services or MultiServices PIC



> **NOTE**: When an AS PIC experiences persistent back pressure as a result of high traffic volume for 3 seconds, the condition triggers an automatic core dump and reboot of the PIC to help clear the blockage. A system log message at level LOG_ERR is generated. This mechanism applies to both Layer 2 and Layer 3 service packages.

RELATED DOCUMENTATION

Services PICs-Overview

Supported Platforms

Services Configuration Procedure

CHAPTER 2

Services Configuration Overview

IN THIS CHAPTER

- Service Sets | 7
- TCP Fast Open | 28
- Service Filters | 39
- Applying Filters and Services to Interfaces | 44
- Configuring Queuing and Scheduling on Inline Services Interface | 47
- Configuring the Address and Domain for Services Interfaces | 49
- Enabling Session Offloading for Multiservices DPCs | 50

Service Sets

IN THIS SECTION

- Understanding Service Sets | 8
- Configuring Service Sets to be Applied to Services Interfaces | 10
- Configuring Service Set Limitations | 15
- Example: Configuring Service Sets | 17
- Configuring Service Interface Pools | 17
- Enabling Services PICs to Accept Multicast Traffic | 17
- Applying Filters and Services to Interfaces | 18
- Examples: Configuring Services Interfaces | 21
- Configuring the Address and Domain for Services Interfaces | 23
- Configuring System Logging for Service Sets | 24
- Configuring Service Rules | 27

Understanding Service Sets

Junos OS enables you to create service sets that define a collection of services to be performed by an Adaptive Services interface (AS) or Multiservices line cards (MS-DPC, MS-MIC, and MS-MPC). You can configure the service set either as an interface-style service set or as a next-hop-style service set.

An interface service set is used as an action modifier across an entire interface. You can use an interface-style service set when you want to apply services to packets passing through an interface.

A next-hop service set is a route-based method of applying a particular service. Only packets destined for a specific next hop are serviced by the creation of explicit static routes. This configuration is useful when services need to be applied to an entire virtual private network (VPN) routing and forwarding (VRF) table, or when routing decisions determine that services need to be performed. When a next-hop service is configured, the service interface is considered to be a two-legged module with one leg configured to be the inside interface (inside the network) and the other configured as the outside interface (outside the network).

In order to avoid packet drop during a service-set deactivate or a service-set delete operation, first bring down the interfaces corresponding to the service-set, wait for sometime , and later deactivate or delete the service-set. However, if the traffic flow is very high, this workaround does not help.

To configure service sets, include the following statements at the `[edit services]` hierarchy level:

```
[edit services]
service-set service-set-name {
    (ids-rules rule-names | ids-rule-sets rule-set-name);
    (ipsec-vpn-rules rule-names | ipsec-vpn-rule-sets rule-set-name);
    max-session-setup-rate max-setup-rate;
    (nat-rules rule-names | nat-rule-sets rule-set-name);
    (pgcp-rules rule-names | pgcp-rule-sets rule-set-name);
    (ptsp-rules rule-names | ptsp-rule-sets rule-set-name);
    (stateful-firewall-rules rule-names | stateful-firewall-rule-sets rule-set-name);
    allow-multicast;
    extension-service service-name {
        provider-specific rules;
    }
    interface-service {
        service-interface interface-name;
    }
    ipsec-vpn-options {
        anti-replay-window-size bits;
        clear-dont-fragment-bit;
        ike-access-profile profile-name;
        local-gateway address;
        no-anti-replay;
        passive-mode-tunneling;
        trusted-ca [ ca-profile-names ];
        tunnel-mtu bytes;
    }
    max-flows number;
    next-hop-service {
        inside-service-interface interface-name.unit-number;
        outside-service-interface interface-name.unit-number;
        service-interface-pool name;
    }
    syslog {
        host hostname {
            services severity-level;
            facility-override facility-name;
            log-prefix prefix-value;
        }
    }
}
adaptive-services-pics {
    traceoptions {
        file filename <files number> <match regex> <size size> <(world-readable | no-world-
readable)>;
        flag flag;
    }
}
logging {
    traceoptions {
        file filename <files number> <match regex> <size size> <(world-readable | no-world-
readable)>;
        flag flag;
    }
}
```

SEE ALSO

Configuring Service Rules | 27

Configuring System Logging for Service Sets | 24

Tracing Services PIC Operations | 35

Configuring Service Sets to be Applied to Services Interfaces

IN THIS SECTION

- Configuring Interface Service Sets | 10
- Configuring Next-Hop Service Sets | 12
- Determining Traffic Direction | 13

You configure a services interface to specify the adaptive services interface on which the service is to be performed. Services interfaces are used with either of the service set types described in the following sections.

Configuring Interface Service Sets

An interface service set is used as an action modifier across an entire interface. To configure the services interface, include the `interface-service` statement at the `[edit services service-set service-set-name]` hierarchy level:

```
[edit services service-set service-set-name]
interface-service {
    service-interface interface-name;
}
```

Only the device name is needed, because the router software manages logical unit numbers automatically. The services interface must be an adaptive services interface for which you have configured `unit 0 family inet` at the `[edit interfaces interface-name` hierarchy level.

When you have defined and grouped the service rules by configuring the service-set definition, you can apply services to one or more interfaces installed on the router. When you apply the service set to an interface, it automatically ensures that packets are directed to the PIC.

To associate a defined service set with an interface, include a `service-set` statement with the `input` or `output` statement at the `[edit interfaces interface-name unit logical-unit-number family inet service]` hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family inet service]
input {
    service-set service-set-name <service-filter filter-name>;
    post-service-filter filter-name;
}
output {
    service-set service-set-name <service-filter filter-name>;
}
```

If a packet is entering the interface, the match direction is `input`. If a packet is leaving the interface, the match direction is `output`. The service set retains the input interface information even after services are applied, so that functions such as filter-class forwarding and destination class usage (DCU) that depend on input interface information continue to work.

You configure the same service set on the input and output sides of the interface. You can optionally include filters associated with each service set to refine the target and additionally process the traffic. If you include the `service-set` statement without a `service-filter` definition, the router software assumes the match condition is true and selects the service set for processing automatically.

NOTE: If you configure service sets with filters, they must be configured on the input and output sides of the interface.

You can include more than one service set definition on each side of the interface. If you include multiple service sets, the router software evaluates them in the order in which they appear in the configuration. The system executes the first service set for which it finds a match in the service filter and ignores the subsequent definitions. A maximum of six service sets can be applied to an interface. When you apply multiple service sets to an interface, you must also configure and apply a service filter to the interface.

An additional statement allows you to specify a filter for processing the traffic after the input service set is executed. To configure this type of filter, include the `post-service-filter` statement at the `[edit interfaces` *`interface-name`* `unit` *`logical-unit-number`* `family inet service input]` hierarchy level:

```
post-service-filter filter-name;
```

The `post-service-filter` statement is not supported when the service interface is on an MS-MIC or MS-MPC.

For an example, see "Example: Configuring Service Sets" on page 17 .

NOTE: With interface-style service sets that are configured with Junos OS extension-provide packages, the traffic fails to get serviced when the ingress interface is part of a VRF instance and the service interface is not part of the same VRF instance.

NOTE: When the MultiServices PIC configured for a service set is either administratively taken offline or undergoes a failure, all the traffic entering the configured interface with an IDP service set would be dropped without notification. To avoid this traffic loss, include the `bypass-traffic-on-pic-failure` statement at the `[edit services service-set` *`service-set-name`* `service-set-options]` hierarchy level. When this statement is configured, the affected packets are forwarded in the event of a MultiServices PIC failure or offlining, as though interface-style services were not configured. This issue applies only to Junos Application Aware (previously known as Dynamic Application Awareness) configurations using IDP service sets. This forwarding feature worked only with the Packet Forwarding Engine (PFE) initially. Starting with Junos OS Release 11.3, the packet-forwarding feature is extended to packets generated by the Routing Engine for bypass service sets as well.

Configuring Next-Hop Service Sets

A next-hop service set is a route-based method of applying a particular service. Only packets destined for a specific next hop are serviced by the creation of explicit static routes. This configuration is useful when services need to be applied to an entire virtual private network (VPN) routing and forwarding (VRF) table, or when routing decisions determine that services need to be performed.

When a next-hop service is configured, the AS or Multiservices PIC is considered to be a two-legged module with one leg configured to be the inside interface (inside the network) and the other configured as the outside interface (outside the network).

NOTE: You can create IFL indexes greater than 8000 only if the interface service set is not configured.

To configure the domain, include the `service-domain` statement at the `[edit interfaces` *`interface-name`* `unit` *`logical-unit-number`*`]` hierarchy level:

```
service-domain (inside | outside);
```

The `service-domain` setting must match the configuration for the next-hop service inside and outside interfaces. To configure the inside and outside interfaces, include the `next-hop-service` statement at the `[edit services service-set` *`service-set-name`*`]` hierarchy level. The interfaces you specify must be logical interfaces on the same AS PIC. You cannot configure `unit 0` for this purpose, and the logical interface you choose must not be used by another service set.

```
next-hop-service {
    inside-service-interface interface-name.unit-number;
    outside-service-interface interface-name.unit-number;
}
```

Traffic on which the service is applied is forced to the inside interface using a static route. For example:

```
routing-options {
    static {
        route 10.1.2.3 next-hop sp-1/1/0.1;
    }
}
```

After the service is applied, traffic exits by way of the outside interface. A lookup is then performed in the Packet Forwarding Engine (PFE) to send the packet out of the AS or Multiservices PIC.

The reverse traffic enters the outside interface, is serviced, and sent to the inside interface. The inside interface forwards the traffic out of the AS or Multiservices PIC.

Determining Traffic Direction

When you configure next-hop service sets, the AS PIC functions as a two-part interface, in which one part is the *inside* interface and the other part is the *outside* interface. The following sequence of actions takes place:

1. To associate the two parts with logical interfaces, you configure two logical interfaces with the `service-domain` statement, one with the `inside` value and one with the `outside` value, to mark them as either an inside or outside service interface.

2. The router forwards the traffic to be serviced to the inside interface, using the next-hop lookup table.

3. After the service is applied, the traffic exits from the outside interface. A route lookup is then performed on the packets to be sent out of the router.

4. When the reverse traffic returns on the outside interface, the applied service is undone; for example, IPsec traffic is decrypted or NAT addresses are unmasked. The serviced packets then emerge on the inside interface, the router performs a route lookup, and the traffic exits the router.

A service rule's match direction, whether input, output, or input/output, is applied with respect to the traffic flow through the AS PIC, not through a specific inside or outside interface.

When a packet is sent to an AS PIC, packet direction information is carried along with it. This is true for both interface style and next-hop style service sets.

Interface Style Service Sets

Packet direction is determined by whether a packet is entering or leaving any Packet Forwarding Engine interface (with respect to the forwarding plane) on which the `interface-service` statement is applied. This is similar to the input and output direction for stateless firewall filters.

The match direction can also depend on the network topology. For example, you might route all the external traffic through one interface that is used to protect the other interfaces on the router, and configure various services on this interface specifically. Alternatively, you might use one interface for priority traffic and configure special services on it, but not care about protecting traffic on the other interfaces.

Next-Hop Style Service Sets

Packet direction is determined by the AS PIC interface used to route packets to the AS PIC. If you use the `inside-interface` statement to route traffic, then the packet direction is `input`. If you use the `outside-interface` statement to direct packets to the AS PIC, then the packet direction is `output`.

The interface to which you apply the service sets affects the match direction. For example, apply the following configuration:

```
sp-1/1/0 unit 1 service-domain inside;
sp-1/1/0 unit 2 service-domain outside;
```

If you configure `match-direction input`, you include the following statements:

```
[edit]
services service-set test1 next-hop-service inside-service-interface sp-1/0/0.1;
services service-set test1 next-hop-service outside-service-interface sp-1/0/0.2;
services ipsec-vpn rule test-ipsec-rule match-direction input;
routing-options static route 10.0.0.0/24 next-hop sp-1/1/0.1;
```

If you configure `match-direction output`, you include the following statements:

```
[edit]
services service-set test2 next-hop-service inside-service-interface sp-1/0/0.1;
services service-set test2 next-hop-service outside-service-interface sp-1/0/0.2;
services ipsec-vpn rule test-ipsec-rule match-direction output;
routing-options static route 10.0.0.0/24 next-hop sp-1/1/0.2;
```

The essential difference between the two configurations is the change in the match direction and the static routes' next hop, pointing to either the AS PIC's inside or outside interface.

SEE ALSO

Example: Configuring Service Sets | 17

Configuring Service Set Limitations

You can set the following limitations on service set capacity:

- You can limit the maximum number of flows allowed per service set. To configure the maximum value, include the `max-flows` statement at the `[edit services service-set service-set-name]` hierarchy level:

  ```
  [edit services service-set service-set-name]
  max-flows number;
  ```

  The `max-flows` statement permits you to assign a single flow limit value. For IDS service sets only, you can specify various types of flow limits with a finer degree of control. For more information, see the description of the `session-limit` statement in "Configuring IDS Rule Sets on an MS-DPC" on page 600 .

  NOTE: When an aggregated multiservices (AMS) interface is configured as the service interface for a service set, the `max-flow` value configured for the service set is applied to each of the member interfaces in the AMS interface. That is, if you have configured 1000 as the `max-flow` value for a service set that uses an AMS interface with four active member interfaces, each of the member interfaces can handle 1000 flows each, resulting in an effective `max-flow` value of 4000.

- You can limit the maximum segment size (MSS) allowed by the Transmission Control Protocol (TCP). To configure the maximum value, include the `tcp-mss` statement at the `[edit services service-set service-set-name]` hierarchy level:

  ```
  [edit services service-set service-set-name]
  tcp-mss number;
  ```

  The TCP protocol negotiates an MSS value during session connection establishment between two peers. The MSS value negotiated is primarily based on the MTU of the interfaces to which the communicating peers are directly connected to. However in the network, due to variation in link MTU on the path taken by the TCP packets, some packets that are still well within the MSS value may be fragmented when the concerned packet's size exceeds the link's MTU.

  If the router receives a TCP packet with the SYN bit and MSS option set and the MSS option specified in the packet is larger than the MSS value specified by the `tcp-mss` statement, the router replaces the MSS value in the packet with the lower value specified by the `tcp-mss` statement. The range for the `tcp-mss` *`mss-value`* parameter is from **536** through **65535**.

  To view statistics of SYN packets received and SYN packets whose MSS value, is modified, issue the `show services service-sets statistics tcp-mss` operational mode command. For more information on this topic, see the Junos OS Administration Library.

- Starting in Junos OS Release 17.1R1, you can limit the session setup rate per service set for an MS-MPC. To configure the maximum setup rate allowed, include the `max-session-setup-rate` statement at the `[edit services service-set service-set-name]` hierarchy level:

  ```
  [edit services service-set service-set-name]
  max-session-setup-rate (number | numberk);
  ```

  The maximum session setup rate is the maximum number of session setups allowed per second. After this rate is reached, any additional session setup attempts are dropped.

  The range for the `max-session-setup-rate` *number* is 1 through 429,496,729. You can also express the setup rate as thousands of sessions by using *number*k. Starting in Junos OS Release 18.4R1, 1k=1000 for the `max-session-setup-rate`. Prior to Junos OS Release 18.4R1, 1k=1024. If you do not include the `max-session-setup-rate` statement, the session setup rate is not limited.

SEE ALSO

Understanding Service Sets | 8

Configuring Service Sets to be Applied to Services Interfaces | 10

Example: Configuring Service Sets

Apply two service sets, `my-input-service-set` and `my-output-service-set`, on an interface-wide basis. All traffic has `my-input-service-set` applied to it. After the service set is applied, additional filtering is done using `my_post_service_input_filter`.

```
[edit interfaces fe-0/1/0]
unit 0 {
    family inet {
        service {
            input {
                service-set my-input-service-set;
                post-service-filter my_post_service_input_filter;
            }
            output {
                service-set my-output-service-set;
            }
        }
    }
}
```

Configuring Service Interface Pools

To configure a service interface pool, include the following statements at the `[edit services service-interface-pools]` hierarchy level:

```
[edit services service-interface-pools]
pool pool-name {
    interface interface-name.unit-number;
}
```

Enabling Services PICs to Accept Multicast Traffic

To allow multicast traffic to be sent to the Adaptive Services or Multiservices PIC, include the `allow-multicast` statement at the `[edit services service-set` *`service-set-name`*`]` hierarchy level. If this statement is not included, multicast traffic is dropped by default. This statement applies only to multicast traffic using a next-hop service set; interface service set configuration is not supported. Only unidirectional flows are created for multicast packets.

SEE ALSO

- Understanding Service Sets | 8
- Configuring Service Sets to be Applied to Services Interfaces | 10
- Example: Configuring Service Sets | 17
- Example: Configuring NAT for Multicast Traffic | 150

Applying Filters and Services to Interfaces

IN THIS SECTION

- Configuring Service Filters | 19

When you have defined and grouped the service rules by configuring the service-set definition, you can apply services to one or more interfaces on the router. To associate a defined service set with an interface, include the `service-set` statement with the `input` or `output` statement at the `[edit interfaces` *`interface-name`* `unit` *`logical-unit-number`* `family inet service]` hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family inet service]
input {
    service-set service-set-name <service-filter filter-name>;
    post-service-filter filter-name;
}
output {
    service-set service-set-name <service-filter filter-name>;
}
```

NOTE: When you enable services on an interface, reverse-path forwarding is not supported. You cannot configure services on the management interface (`fxp0`) or the loopback interface (`lo0`).

You can configure different service sets on the input and output sides of the interface. However, for service sets with bidirectional service rules, you must include the same service set definition in both the `input` and `output` statements. Any service set you include in the `service` statement must be configured with the `interface-service` statement at the `[edit services service-set` *`service-set-name`*`]` hierarchy level; for more information, see "Configuring Service Sets to be Applied to Services Interfaces" on page 10 .

NOTE: If you configure an interface with an input firewall filter that includes a reject action and with a service set that includes stateful firewall rules, the router executes the input firewall filter before the stateful firewall rules are run on the packet. As a result, when the Packet Forwarding Engine sends an Internet Control Message Protocol (ICMP) error message out through the interface, the stateful firewall rules might drop the packet because it was not seen in the input direction.

Possible workarounds are to include a forwarding-table filter to perform the reject action, because this type of filter is executed after the stateful firewall in the input direction, or to include an output service filter to prevent the locally generated ICMP packets from going to the stateful firewall service.

Configuring Service Filters

You can optionally include filters associated with each service set to refine the target and additionally process the traffic. If you include the `service-set` statement without a `service-filter` definition, the router software assumes that the match condition is true and selects the service set for processing automatically.

To configure service filters, include the `firewall` statement at the `[edit]` hierarchy level:

```
firewall {
    family inet {
        service-filter filter-name {
            term term-name {
                from {
                    match-conditions;
                }
                then {
                    action;
                    action-modifiers;
                }
            }
        }
    }
}
```

NOTE: You must specify inet as the address family to configure a service filter.

You configure service filters in a similar way to firewall filters. Service filters have the same match conditions as firewall filters, but the following specific actions:

- count—Add the packet to a counter total.
- log—Log the packet.
- port-mirror—Port-mirror the packet.
- sample—Sample the packet.
- service—Forward the packet for service processing.
- skip—Omit the packet from service processing.

For more information about configuring firewall filters, see the Routing Policies, Firewall Filters, and Traffic Policers User Guide.

You can also include more than one service set definition on each side of the interface. If you include multiple service sets, the router software evaluates them in the order specified in the configuration. It executes the first service set for which it finds a match in the service filter and ignores the subsequent definitions.

An additional statement allows you to specify a filter for processing the traffic after the input service set is executed. To configure this type of filter, include the post-service-filter statement at the [edit interfaces *interface-name* unit *logical-unit-number* family inet service input] hierarchy level:

```
post-service-filter filter-name;
```

NOTE: The software performs postservice filtering only when it has selected and executed a service set. If the traffic does not meet the match criteria for any of the configured service sets, the postservice filter is ignored. The post-service-filter statement is not supported when the service interface is on an MS-MIC or MS-MPC.

For an example of applying a service set to an interface, see "Examples: Configuring Services Interfaces" on page 21 .

For more information on applying filters to interfaces, see the Junos OS Network Interfaces Library for Routing Devices. For general information on filters, see the Routing Policies, Firewall Filters, and Traffic Policers User Guide.

NOTE: After NAT processing is applied to packets, they are not subject to output service filters. The service filters affect only untranslated traffic.

Examples: Configuring Services Interfaces

Apply the my-service-set service set on an interface-wide basis. All traffic that is accepted by my_input_filter has my-input-service-set applied to it. After the service set is applied, additional filtering is done using the my_post_service_input_filter filter.

```
[edit interfaces fe-0/1/0]
unit 0 {
    family inet {
        filter {
            input my_input_filter;
            output my_output_filter;
        }
        service {
            input {
                service-set my-input-service-set;
                post-service-filter my_post_service_input_filter;
            }
            output {
                service-set my-output-service-set;
            }
        }
    }
}
```

Configure two redundancy interfaces, rsp0 and rsp1, and associated services.

```
[edit interfaces]
rsp0 {
    redundancy-options {
        primary sp-0/0/0;
        secondary sp-1/3/0;
    }
    unit 0 {
        family inet;
    }
    unit 30 {
        family inet;
        service-domain inside;
    }
    unit 31 {
        family inet;
        service-domain outside;
    }
}
rsp1 {
    redundancy-options {
        primary sp-0/1/0;
        secondary sp-1/3/0;
    }
    unit 0 {
        family inet;
    }
    unit 20 {
        family inet;
        service-domain inside;
    }
    unit 21 {
        family inet;
        service-domain outside;
    }
}
[edit services]
service-set null-sfw-with-nat {
    stateful-firewall-rules allow-all;
    nat-rules rule1;
    next-hop-service {
        inside-service-interface rsp0.30;
        outside-service-interface rsp0.31;
    }
}
[edit routing-instances]
vpna {
    interface rsp0.0;
}
```

SEE ALSO

Configuring System Logging for Services Interfaces

Applying Filters and Services to Interfaces | 18

Example: Configuring an Aggregated Multiservices Interface (AMS) | 1025

Configuring the Address and Domain for Services Interfaces

On the AS or Multiservices PIC, you configure a source address for system log messages by including the `address` statement at the `[edit interfaces interface-name unit logical-unit-number family inet]` hierarchy level:

```
address address {
    ...
}
```

Assign an IP address to the interface by configuring the *address* value. The AS or Multiservices PIC generally supports only IP version 4 (IPv4) addresses configured using the `family inet` statement, but IPsec services support IP version 6 (IPv6) addresses as well, configured using the `family inet6` statement.

NOTE: If you configure the same address on multiple interfaces in the same routing instance, Junos OS uses only the first configuration, the remaining address configurations are ignored and can leave interfaces without an address. Interfaces that do not have an assigned address cannot be used as a donor interface for an unnumbered Ethernet interface.

For example, in the following configuration the address configuration of interface xe-0/0/1.0 is ignored:

```
interfaces {
  xe-0/0/0 {
    unit 0 {
      family inet {
        address 192.168.1.1/24;
      }
    }
  }
  xe-0/0/1 {
    unit 0 {
     family inet {
       address 192.168.1.1/24;
     }
  }
}
```

For more information on configuring the same address on multiple interfaces, see *Configuring the Interface Address*.

For information on other addressing properties you can configure that are not specific to service interfaces, see the Junos OS Network Interfaces Library for Routing Devices.

The `service-domain` statement specifies whether the interface is used within the network or to communicate with remote devices. The software uses this setting to determine which default stateful firewall rules to apply, and to determine the default direction for service rules. To configure the domain, include the `service-domain` statement at the `[edit interfaces interface-name unit logical-unit-number]` hierarchy level:

```
service-domain (inside | outside);
```

If you are configuring the interface in a next-hop service-set definition, the `service-domain` setting must match the configuration for the `inside-service-interface` and `outside-service-interface` statements; for more information, see "Configuring Service Sets to be Applied to Services Interfaces" on page 10 .

SEE ALSO

Configuring Default Timeout Settings for Services Interfaces

Example: Configuring an Aggregated Multiservices Interface (AMS) | **1025**

Configuring System Logging for Service Sets

You specify properties that control how system log messages are generated for the service set. These values override the values configured at the `[edit interfaces interface-name services-options]` hierarchy level.

To configure service-set-specific system logging values, include the `syslog` statement at the `[edit services service-set service-set-name]` hierarchy level:

```
syslog {
    host hostname {
        class class-name
        facility-override facility-name;
        log-prefix prefix-value;
        port port-number
        services severity-level;
        source-address source-address
    }
}
```

Configure the host statement with a hostname or an IP address that specifies the system log target server. The hostname local directs system log messages to the Routing Engine. For external system log servers, the hostname must be reachable from the same routing instance to which the initial data packet (that triggered session establishment) is delivered. You can specify only one system logging hostname. The source-address parameter is supported on the ms, rms, and mams interfaces.

Starting in Junos OS Release 17.4R1, you can configure up to a maximum of four system log servers (combination of local system log hosts and remote system log collectors) for each service set under [edit services service-set *service-set-name*] hierarchy level.

NOTE: Junos OS does not support the exporting of system log messages to an external system log server through the fxp.0 interface; this is because the high transmission rate of system log messages and the limited bandwidth of the fxp.0 interface can cause several problems. The external system log server must be reachable through a routable interface.

Table 1 on page 25 lists the severity levels that you can specify in configuration statements at the [edit services service-set *service-set-name* syslog host *hostname*] hierarchy level. The levels from emergency through info are in order from highest severity (greatest effect on functioning) to lowest.

Table 1: System Log Message Severity Levels

Severity Level	Description
any	Includes all severity levels
emergency	System panic or other condition that causes the router to stop functioning
alert	Conditions that require immediate correction, such as a corrupted system database
critical	Critical conditions, such as hard drive errors

Table 1: System Log Message Severity Levels *(Continued)*

Severity Level	Description
error	Error conditions that generally have less serious consequences than errors in the emergency, alert, and critical levels
warning	Conditions that warrant monitoring
notice	Conditions that are not errors but might warrant special handling
info	Events or non-error conditions of interest

We recommend setting the system logging severity level to `error` during normal operation. To monitor PIC resource usage, set the level to `warning`. To gather information about an intrusion attack when an intrusion detection system error is detected, set the level to `notice` for a specific service set. To debug a configuration or log NAT functionality, set the level to `info`.

For more information about system log messages, see the System Log Explorer.

To select the class of messages to be logged to the specified system log host, include the `class` statement at the `[edit services service-set` *`service-set-name`* `syslog host` *`hostname`*`]` hierarchy level:

```
class class-name;
```

To use one particular facility code for all logging to the specified system log host, include the `facility-override` statement at the `[edit services service-set` *`service-set-name`* `syslog host` *`hostname`*`]` hierarchy level:

```
facility-override facility-name;
```

The supported facilities are: `authorization`, `daemon`, `ftp`, `kernel`, `user`, and `local0` through `local7`.

To specify a text prefix for all logging to this system log host, include the `log-prefix` statement at the `[edit services service-set` *`service-set-name`* `syslog host` *`hostname`*`]` hierarchy level:

```
log-prefix prefix-value;
```

SEE ALSO

> Tracing Services PIC Operations | 35

Configuring Service Rules

You specify the collection of rules and rule sets that constitute the service set. The router performs rule sets in the order in which they appear in the configuration. You can include only one rule set for each service type. You configure the rule names and content for each service type at the `[edit services name]` hierarchy level for each type:

- You configure intrusion detection service (IDS) rules at the `[edit services ids]` hierarchy level; for more information, see "Configuring IDS Rules on an MS-DPC" on page 590 for MS-DPC cards and "Configuring Protection Against Network Attacks on an MS-MPC" on page 610 for MS-MPC cards.
- You configure IP Security (IPsec) rules at the `[edit services ipsec-vpn]` hierarchy level; for more information, see "Understanding Junos VPN Site Secure" on page 623 ..
- You configure Network Address Translation (NAT) rules at the `[edit services nat]` hierarchy level; for more information, see "Junos Address Aware Network Addressing Overview" on page 53 ..
- You configure packet-triggered subscribers and policy control (PTSP) rules at the `[edit services ptsp]` hierarchy level; for more information, see Configuring PTSP Service Rules.
- You configure softwire rules for DS-Lite or 6rd softwires at the `[edit services softwire]` hierarchy level; for more information, see "Configuring Softwire Rules" on page 399 .
- You configure stateful firewall rules at the `[edit services stateful-firewall]` hierarchy level; for more information, see "Configuring Stateful Firewall Rules" on page 550 .

To configure the rules and rule sets that constitute a service set, include the following statements at the `[edit services service-set service-set-name]` hierarchy level:

```
([ ids-rules rule-names ] |ids-rule-sets rule-set-name);
([ ipsec-vpn-rules rule-names ] | ipsec-vpn-rule-sets rule-set-name);
([ nat-rules rule-names ] | nat-rule-sets rule-set-name);
([ pgcp-rules rule-names] | pgcp-rule-sets rule-set-name);
([softwire-rules rule-names] | softwire-rule-sets rule-set-name);
([ stateful-firewall-rules rule-names ] | stateful-firewall-rule-sets rule-set-name);
```

For each service type, you can include one or more individual rules, or one rule set.

If you configure a service set with IPsec rules, it must not contain rules for any other services. You can, however, configure another service set containing rules for the other services and apply both service sets to the same interface.

NOTE: You can also include Junos Application Aware (previously known as Dynamic Application Awareness) functionality within service sets. To do this, you must include an `idp-profile` statement at the `[edit services service-set]` hierarchy level, along with application identification (APPID) rules, and, as appropriate, application-aware access list (AACL) rules and a `policy-decision-statistics-profile`. Only one service sets can be applied to a single interface when Junos Application Aware functionality is used. For more information, see "Configuring IDS Rules on an MS-DPC" on page 590 , *APPID Overview*, and Application Aware Services Interfaces User Guide for Routing Devices.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
18.4R1	Starting in Junos OS Release 18.4R1, 1k=1000 for the `max-session-setup-rate`.
17.1R1	Starting in Junos OS Release 17.1R1, you can limit the session setup rate per service set for an MS-MPC.

TCP Fast Open

IN THIS SECTION

- Exchanging Data More Efficiently Using TCP Fast Open | 28
- Configuring TFO | 30
- Configuring Fragmentation Control for MS-DPC and MS-PIC Service Interfaces | 34
- Tracing Services PIC Operations | 35

Exchanging Data More Efficiently Using TCP Fast Open

TCP Fast Open (TFO) is an update to TCP that saves up to one full round-trip time (RTT) over the standard three-way connection handshake during a TCP session. TFO support is for MS-MPC and MS-MIC.

The standard three-way connection handshake involves three sets of send and receive messages between two hosts and the following exchange of SYN (synchronize) and ACK (acknowledgement) packets:

1. Host A sends a TCP SYN packet to Host B. Host B receives it.

2. Host B sends a SYN-ACK packet to Host A. Host A receives it.

3. Host A sends an ACK packet to Host B. Host B receives it.

In standard TCP, although data can be carried in SYN packets, this data cannot be delivered until the three-way handshake is completed. TFO removes this constraint and allows data in SYN packets to be delivered to the application, yielding significant latency improvement.

The key component of TFO is the Fast Open Cookie (cookie), which is a Message Authentication Code (MAC) tag generated by the server. The client requests a cookie in one regular TCP connection, then uses it for future TCP connections to exchange data during the handshake.

The TFO option is used to request or to send a TFO cookie. When a cookie is not present or is empty, the option is used by the client to request a cookie from the server. When the cookie is present, the option is used to pass the cookie from the server to the client or from the client back to the server.

The following list outlines how the client requests a TFO cookie:

1. The client sends a SYN with a TFO option that has the cookie field empty.

2. The server generates a cookie and sends it through the TFO option of a SYN-ACK packet.

3. The client caches the cookie for future TFO connections.

Thereafter, the two devices perform a TFO exchange:

1. The client sends a SYN with data and the cookie in the TFO option.

2. The server validates the cookie:

   - If the cookie is valid, the server sends a SYN-ACK acknowledging both the SYN and the data.

     The server then delivers the data to the application.

   - Otherwise, the server drops the data and sends a SYN-ACK acknowledging only the SYN sequence number.

The rest of the connection proceeds like a normal TCP connection. The client can repeat many TFO operations once it acquires a cookie (until the cookie is expired by the server). Thus, TFO is useful for applications in which the same client reconnects to the same server multiple times and exchanges data.

SEE ALSO

tcp-fast-open

Configuring TFO

IN THIS SECTION

- Three Modes for TFO | 30
- Using NAT and TFO | 33

In this topic, the three modes of TCP Fast Open (TFO) are described and examples given. The case of using NAT with TFO is also covered.

Three Modes for TFO

No configuration is required to use TFO. TFO is enabled by default. In default mode, all TFO packets are forwarded by the service PIC. Besides the default, there are two other modes for TFO that you configure through the CLI:

- Drop TFO—If this mode is set, no TFO packets are forwarded.
- Disable TFO—If this mode is set, any SYN or SYN ACK packet carrying TFO, data, or both, will be stripped of the TFO and the data before being forwarded.

The TFO option is enabled per service set. The service set can be either a next-hop service set or an interface-style service set. Following is an example interface-style service set configuration:

```
[edit]
services {
    service-set ss2 {
        stateful-firewall-rules sfw_rule;
        interface-service {
            service-interface ms-2/3/0;
        }
    }
    stateful-firewall {
        rule sfw_rule {
            match-direction input-output;
            term 0 {
                from {
                    source-address {
                        any-ipv4;
                    }
                    destination-address {
                        any-ipv4;
                    }
                    then {
                        accept;
                    }
                }
            }
        }
    }
}
```

In this instance, TFO is enabled by default (no TFO configuration). The output for the `show services service-sets statistics tcp` command is as follows:

```
user@host> show services service-sets statistics tcp

Interface: ms-2/3/0
  Service set: ss2
    TCP open/close statistics:
      TCP first packet non-syn: 0
      TCP first packet reset: 0
      TCP first packet FIN: 0
      TCP non syn discard: 0
      TCP extension alloc fail: 0
      TFO SYN with cookie request: 1
      TFO SYN with cookie: 0
      TFO SYN ACK with cookie: 0
      TFO packets forwarded: 0
      TFO packets dropped: 1
      TFO packets stripped: 0
```

If you drop TFO enabled packets, you have the following configuration and output:

```
[edit]
services {
    service-set ss2 {
        service-set-options {
            tcp-fast-open drop;
        }
        stateful-firewall-rules sfw_rule;
        interface-service {
            service-interface ms-2/3/0;
        }
    }
}
```

```
user@host> show services service-sets statistics tcp

Interface: ms-2/3/0
  Service set: ss2
    TCP open/close statistics:
      TCP first packet non-syn: 0
      TCP first packet reset: 0
      TCP first packet FIN: 0
      TCP non syn discard: 0
      TCP extension alloc fail: 0
      TFO SYN with cookie request: 1
      TFO SYN with cookie: 0
      TFO SYN ACK with cookie: 0
      TFO packets forwarded: 0
      TFO packets dropped: 1
      TFO packets stripped: 0
```

If you strip the TFO option, the configuration and output change accordingly:

```
[edit]
services {
    service-set ss2 {
        service-set-options {
            tcp-fast-open disabled;
        }
        stateful-firewall-rules sfw_rule;
        interface-service {
            service-interface ms-2/3/0;
        }
    }
}
```

```
user@host> show services service-sets statistics tcp

Interface: ms-2/3/0
  Service set: ss2
    TCP open/close statistics:
      TCP first packet non-syn: 0
      TCP first packet reset: 0
      TCP first packet FIN: 0
      TCP non syn discard: 0
      TCP extension alloc fail: 0
      TFO SYN with cookie request: 1
      TFO SYN with cookie: 0
      TFO SYN ACK with cookie: 0
      TFO packets forwarded: 0
      TFO packets dropped: 0
      TFO packets stripped: 1
```

Using NAT and TFO

If NAT is configured in the service set and you are using TFO, you should configure address-pooling paired (APP). APP allows a private IP address to be mapped to the same public IP address from a NAT pool for all its sessions.

If you do not configure APP, NAT can give a different IP address to the client from the same NAT pool than the one it sent to the server before. The server does not recognize the IP address, drops the TFO option, and replies with SYN ACK and the data the client sent is not acknowledged. Therefore, even though the connection is successful and no packet is lost, the benefit of TFO is lost. But if client comes back with the same IP address, the server recognizes it and acknowledges the data. Therefore, always enable APP with a high mapping timeout value with TFO.

To configure APP:

1. Configure APP:

   ```
   set services nat rule rule-name term term-name then translated address-pooling paired
   ```

2. Configure a high mapping timeout value:

```
set services nat pool nat-pool-name mapping-timeout seconds
```

RELATED DOCUMENTATION

> *tcp-fast-open*

Configuring Fragmentation Control for MS-DPC and MS-PIC Service Interfaces

Two configuration options are available to prevent excessive consumption of computational CPU cycles on a services PIC caused by the handling of large numbers of fragmented packets. Such fragment handling can be exploited in DOS attacks. The `fragment-limit` option establishes a maximum number of fragments for a packet. When this number is exceeded, the packet is dropped. The `reassembly-timeout` specifies the maximum time from the receipt of the first and latest fragments in a packet. When the number is exceeded, the packet is dropped.

To configure fragmentation control for MS-DPC and MS-PIC service interfaces:

1. In configuration mode, go to the `[edit interfaces` *`interface-name`* `services-options` hierarchy level.

```
edit interfaces interface-name services-options
```

2. Configure the fragment limit.

```
[ edit services interface-name services-options]
set fragment-limit number-of-fragments
```

3. Configure the reassembly timeout.

```
[ edit services interface-name services-options]
set reassembly-timeout number-of-fragments
```

Tracing Services PIC Operations

IN THIS SECTION

- Configuring the Adaptive Services Log Filename | 36
- Configuring the Number and Size of Adaptive Services Log Files | 36
- Configuring Access to the Log File | 36
- Configuring a Regular Expression for Lines to Be Logged | 37
- Configuring the Trace Operations | 37

Tracing operations track all adaptive services operations and record them in a log file. The logged error descriptions provide detailed information to help you solve problems faster.

By default, no events are traced. If you include the `traceoptions` statement at the `[edit services adaptive-services-pics]` or `[edit services logging]` hierarchy level, the default tracing behavior is the following:

- Important events are logged in a file called `serviced` located in the **/var/log** directory.
- When the file **serviced** reaches 128 kilobytes (KB), it is renamed **serviced.0**, then **serviced.2**, and so on, until there are three trace files. Then the oldest trace file (**serviced.2**) is overwritten. (For more information about how log files are created, see the System Log Explorer.)
- Log files can be accessed only by the user who configures the tracing operation.

You cannot change the directory (**/var/log**) in which trace files are located. However, you can customize the other trace file settings by including the following statements:

```
file filename <files number> <match regular-expression> <size size> <world-readable | no-world-
readable>;
flag {
    all;
    command-queued;
    config;
    handshake;
    init;
    interfaces;
    mib;
    removed-client;
    show;
}
```

You include these statements at the `[edit services adaptive-services-pics traceoptions]` or `[edit services logging traceoptions]` hierarchy level.

These statements are described in the following sections:

Configuring the Adaptive Services Log Filename

By default, the name of the file that records trace output is **serviced**. You can specify a different name by including the `file` statement at the `[edit services adaptive-services-pics traceoptions]` or `[edit services logging traceoptions]` hierarchy level:

```
file filename;
```

Configuring the Number and Size of Adaptive Services Log Files

By default, when the trace file reaches 128 kilobytes (KB) in size, it is renamed ***filename*.0**, then ***filename*.1**, and so on, until there are three trace files. Then the oldest trace file (***filename*.2**) is overwritten.

You can configure the limits on the number and size of trace files by including the following statements at the `[edit services adaptive-services-pics traceoptions]` or `[edit services logging traceoptions]` hierarchy level:

```
file <filename> files number size size;
```

For example, set the maximum file size to 2 MB, and the maximum number of files to 20. When the file that receives the output of the tracing operation (***filename***) reaches 2 MB, ***filename*** is renamed ***filename*.0**, and a new file called ***filename*** is created. When the new ***filename*** reaches 2 MB, ***filename*.0** is renamed ***filename*.1** and ***filename*** is renamed ***filename*.0**. This process repeats until there are 20 trace files. Then the oldest file (***filename*.19**) is overwritten by the newest file (***filename*.0**).

The number of files can be from 2 through 1000 files. The file size of each file can be from 10 KB through 1 gigabyte (GB).

Configuring Access to the Log File

By default, log files can be accessed only by the user who configures the tracing operation.

To specify that any user can read all log files, include the `file world-readable` statement at the `[edit services adaptive-services-pics traceoptions]` or `[edit services logging traceoptions]` hierarchy level:

```
file <filename> world-readable;
```

To explicitly set the default behavior, include the `file no-world-readable` statement at the `[edit services adaptive-services-pics traceoptions]` or `[edit services logging traceoptions]` hierarchy level:

```
file <filename> no-world-readable;
```

Configuring a Regular Expression for Lines to Be Logged

By default, the trace operation output includes all lines relevant to the logged events.

You can refine the output by including the `match` statement at the `[edit services adaptive-services-pics traceoptions file filename]` or `[edit services logging traceoptions]` hierarchy level and specifying a regular expression (regex) to be matched:

```
file <filename> match regular-expression;
```

Configuring the Trace Operations

By default, if the `traceoptions` configuration is present, only important events are logged. You can configure the trace operations to be logged by including the following statements at the `[edit services adaptive-services-pics traceoptions]` or `[edit services logging traceoptions]` hierarchy level:

```
flag {
    all;
    configuration;
    routing-protocol;
    routing-socket;
    snmp;
}
```

Table 2 on page 38 describes the meaning of the adaptive services tracing flags.

Table 2: Adaptive Services Tracing Flags

Flag	Description	Default Setting
all	Trace all operations.	Off
command-queued	Trace command enqueue events.	Off
config	Log reading of the configuration at the [edit services] hierarchy level.	Off
handshake	Trace handshake events.	Off
init	Trace initialization events.	Off
interfaces	Trace interface events.	Off
mib	Trace GGSN SNMP MIB events.	Off
removed-client	Trace client cleanup events.	Off
show	Trace CLI command servicing.	Off

To display the end of the log, issue the `show log serviced | last` operational mode command:

```
[edit]
user@host# run show log serviced | last
```

Service Filters

IN THIS SECTION

- Service Filters in ACX Series | 39
- Guidelines for Applying Service Filters | 40
- Service Filter Match Conditions for IPv4 Traffic | 42
- Service Filter Actions | 43

Service Filters in ACX Series

When you apply a service set to the traffic at an inline services interface, you can optionally use service filters to refine the target of the set of services and also to process traffic. Service filters enable you to manipulate traffic by performing packet filtering to a defined set of services on an inline services interface before the traffic is delivered to its destination. In ACX Series routers, you can apply a service filter to traffic before packets are accepted for input service processing.

NOTE: In ACX Series routers, the `service-set` filters are implemented using ternary content addressable memory (TCAM) space. The allocated TCAM space is shared by the bridge family filter. The same space is shared by the NNI-Address-Overload-Reverse filter (for each service set that is configured with address overloading, the internal filters are configured for the given overloaded IP address and the port range to redirect the matched reverse-nat (public to private) traffic to the service). From a scaling perspective, the allocated 124 hardware TCAM entries are shared by these features and the allocation of TCAM entries works on a first-come-first-serve basis mode.

SEE ALSO

Network Address Translation Overview on ACX Series | 120

Network Address Port Translation Overview | 121

Enabling Inline Services Interface on ACX Series | 124

Understanding Service Sets

Network Address Translation Address Overload in ACX Series | 122

CoS for NAT Services on ACX Series Routers

Network Address Translation Constraints on ACX | 123
Configuring Address Pools for Network Address Port Translation (NAPT) Overview
Configuring Service Sets to Be Applied to Services Interfaces

Guidelines for Applying Service Filters

IN THIS SECTION

- Restrictions for Inline Services Interfaces | 40
- Statement Hierarchy for Applying Service Filters | 40
- Associating Service Rules with Inline Services Interfaces | 41
- Filtering Traffic Before Accepting Packets for Service Processing | 41

This topic covers the following information:

Restrictions for Inline Services Interfaces

You can apply a service filter to IPv4 traffic associated with a service set at an *inline services interface* only.

ACX Series routers do not support post-service filters.

Statement Hierarchy for Applying Service Filters

You can enable packet filtering of IPv4 traffic before a packet is accepted for input service processing. To do this, apply a service filter to the inline services interface input in conjunction with an interface service set.

The following configuration shows the hierarchy levels at which you can apply the service filters to inline services interfaces:

```
[edit]
interfaces {
    interface-name {
        unit unit-number {
            family (inet | inet6) {
                service {
                    input {
                        service-set service-set-name service-filter service-filter-name;
                    }
                    output {
                        [ service-set service-set-name <service-filter filter-name> ];
                    }
                }
            }
        }
    }
}
```

Associating Service Rules with Inline Services Interfaces

To define and group the service rules be applied to an inline services interface, you define an *interface service set* by including the `service-set` *`service-set-name`* statement at the `[edit services]` hierarchy level.

To apply an interface service set to the input of an inline services interface, you include the **service-set** ***service-set-name*** at the following hierarchy levels:

- `[edit interfaces` *`interface-name`* `unit` *`unit-number`* `input]`

Filtering Traffic Before Accepting Packets for Service Processing

To filter IPv4 traffic before accepting packets for input service processing, include the **service-set** ***service-set-name*** **service-filter** ***service-filter-name*** at the following hierarchy level:

- `[edit interfaces` *`interface-name`* `unit` *`unit-number`* `family inet service input]`

For the ***service-set-name***, specify a service set configured at the `[edit services service-set]` hierarchy level.

The service set retains the input interface information even after services are applied, so that functions such as filter-class forwarding that depend on input interface information continue to work.

The following requirements apply to filtering inbound or outbound traffic before accepting packets for service processing:

- You configure the same service set on the input and output sides of the interface.
- If you include the `service-set` statement without an optional **service-filter** definition, Junos OS assumes that the match condition is true and selects the service set for processing automatically.
- The service filter is applied only if a service set is configured and selected.

SEE ALSO

Enabling Inline Services Interface on ACX Series | 124

Service Filter Match Conditions for IPv4 Traffic

In ACX Series, service filters support only a subset of the stateless firewall filter match conditions for IPv4 traffic. Table 3 on page 42 describes the service filter match conditions.

Table 3: Service Filter Match Conditions for IPv4 Traffic

Match Condition	Description	Protocol Families
destination-address *address*	Match the IP destination address field.	**family inet**
destination-port *number*	Match the UDP or TCP destination port field. You cannot specify both the **port** and **destination-port** match conditions in the same term. If you configure this match condition for IPv4 traffic, we recommend that you also configure the **protocol udp** or **protocol tcp** match statement in the same term to specify which protocol is being used on the port.	**family inet**
ip-options *values*	Match the 8-bit IP option field, if present, to the specified value or list of values.	**family inet**
protocol *number*	Match the IP protocol type field.	**family inet**
source-address *address*	Match the IP source address.	**family inet**
source-port *number*	Match the UDP or TCP source port field. If you configure this match condition for IPv4 traffic, we recommend that you also configure the **protocol udp** or **protocol tcp** match statement in the same term to specify which protocol is being used on the port.	**family inet**

Table 3: Service Filter Match Conditions for IPv4 Traffic *(Continued)*

Match Condition	Description	Protocol Families
tcp-flags *value*	Match one or more of the low-order 6 bits in the 8-bit TCP flags field in the TCP header. If you configure this match condition for IPv4 traffic, we recommend that you also configure the **protocol tcp** match statement in the same term to specify that the TCP protocol is being used on the port.	**family inet**

SEE ALSO

Enabling Inline Services Interface on ACX Series | 124

Configuring Queuing and Scheduling on Inline Services Interface | 47

Service Filter Actions

ACX Series support different sets of terminating and nonterminating actions that you can configure in a service filter term.

NOTE: Service filters do not support the **next term** action.

Table 4 on page 43 describes the terminating actions you can configure in a service filter term.

Table 4: Terminating Actions for Service Filters

Terminating Action	Description	Protocol Families
service	Direct the packet to service processing.	**inet**

Table 5 on page 44 describes the nonterminating actions you can configure in a service filter term.

Table 5: Nonterminating Actions for Service Filters

Nonterminating Action	Description	Protocol Families
accept	Accept the packet.	**inet**
count *counter-name*	Count the packet in the named counter.	**inet**
log	Log the packet header information in a buffer within the Packet Forwarding Engine. You can access this information by issuing the show firewall log command at the command-line interface (CLI).	**inet**
port-mirror	Port-mirror the packet based on the specified family.	**inet**

SEE ALSO

Enabling Inline Services Interface on ACX Series | 124

Applying Filters and Services to Interfaces

IN THIS SECTION

- Configuring Service Filters | 45

When you have defined and grouped the service rules by configuring the service-set definition, you can apply services to one or more interfaces on the router. To associate a defined service set with an interface, include the service-set statement with the input or output statement at the [edit interfaces *interface-name* unit *logical-unit-number* family inet service] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family inet service]
input {
    service-set service-set-name <service-filter filter-name>;
    post-service-filter filter-name;
}
output {
    service-set service-set-name <service-filter filter-name>;
}
```

NOTE: When you enable services on an interface, reverse-path forwarding is not supported. You cannot configure services on the management interface (`fxp0`) or the loopback interface (`lo0`).

You can configure different service sets on the input and output sides of the interface. However, for service sets with bidirectional service rules, you must include the same service set definition in both the `input` and `output` statements. Any service set you include in the `service` statement must be configured with the `interface-service` statement at the `[edit services service-set *service-set-name*]` hierarchy level; for more information, see "Configuring Service Sets to be Applied to Services Interfaces" on page 10 .

NOTE: If you configure an interface with an input firewall filter that includes a reject action and with a service set that includes stateful firewall rules, the router executes the input firewall filter before the stateful firewall rules are run on the packet. As a result, when the Packet Forwarding Engine sends an Internet Control Message Protocol (ICMP) error message out through the interface, the stateful firewall rules might drop the packet because it was not seen in the input direction.

Possible workarounds are to include a forwarding-table filter to perform the reject action, because this type of filter is executed after the stateful firewall in the input direction, or to include an output service filter to prevent the locally generated ICMP packets from going to the stateful firewall service.

Configuring Service Filters

You can optionally include filters associated with each service set to refine the target and additionally process the traffic. If you include the `service-set` statement without a `service-filter` definition, the router software assumes that the match condition is true and selects the service set for processing automatically.

To configure service filters, include the `firewall` statement at the `[edit]` hierarchy level:

```
firewall {
    family inet {
        service-filter filter-name {
            term term-name {
                from {
                    match-conditions;
                }
                then {
                    action;
                    action-modifiers;
                }
            }
        }
    }
}
```

NOTE: You must specify `inet` as the address family to configure a service filter.

You configure service filters in a similar way to firewall filters. Service filters have the same match conditions as firewall filters, but the following specific actions:

- `count`—Add the packet to a counter total.
- `log`—Log the packet.
- `port-mirror`—Port-mirror the packet.
- `sample`—Sample the packet.
- `service`—Forward the packet for service processing.
- `skip`—Omit the packet from service processing.

For more information about configuring firewall filters, see the Routing Policies, Firewall Filters, and Traffic Policers User Guide.

You can also include more than one service set definition on each side of the interface. If you include multiple service sets, the router software evaluates them in the order specified in the configuration. It executes the first service set for which it finds a match in the service filter and ignores the subsequent definitions.

An additional statement allows you to specify a filter for processing the traffic after the input service set is executed. To configure this type of filter, include the `post-service-filter` statement at the `[edit interfaces` *`interface-name`* `unit` *`logical-unit-number`* `family inet service input]` hierarchy level:

```
post-service-filter filter-name;
```

NOTE: The software performs postservice filtering only when it has selected and executed a service set. If the traffic does not meet the match criteria for any of the configured service sets, the postservice filter is ignored. The `post-service-filter` statement is not supported when the service interface is on an MS-MIC or MS-MPC.

For an example of applying a service set to an interface, see "Examples: Configuring Services Interfaces" on page 21 .

For more information on applying filters to interfaces, see the Junos OS Network Interfaces Library for Routing Devices. For general information on filters, see the Routing Policies, Firewall Filters, and Traffic Policers User Guide.

NOTE: After NAT processing is applied to packets, they are not subject to output service filters. The service filters affect only untranslated traffic.

Configuring Queuing and Scheduling on Inline Services Interface

To configure queuing and scheduling on an inline services interface, you need to include `scheduler-map` statement at the `[edit class-of-services interfaces si-/0/0/0]` hierarchy level.

```
[edit class-of-service]
scheduler-maps <scheduler-map-name>;
interfaces si-0/0/0; {
scheduler-map <scheduler-map-name>;
}
```

The `queue-number` *7* of the inline services interface has *strict-high* priority because the timing packets received by ACX Series routers gets assigned to this queue. You can explicitly override this strict-high priority by assigning an explicit scheduler for `queue-number 7` in the `scheduler-map` statement attached to inline services interface as shown below:

```
[edit class-of-service]
forwarding-classes {
    class <class-name> queue-number 7;
}
interfaces {
    si-0/0/0{
    scheduler-map scheduler-map-name;
    }
}
scheduler-maps {
    <map-name> {
    forwarding-class <class-name> scheduler <scheduler-name>;
    }
}
schedulers {
    <scheduler-name> {
    priority low ;
    }
}
```

The following are the CoS limitations for inline services:

- Inline services packets classified with packet loss priority as *medium-high* in the ingress path are treated as *high* on the egress path.
- When both timing and NAT services are enabled on the router, you should not classify NAT traffic into a forwarding class mapped with `queue-number 7`, because if you do so, the performance of timing services can degrade.
- If a scheduler with `queue-number 7` in the `scheduler-map` statement is attached to an inline services interface, then the scheduler should be configured with *strict* priority, else the timing performance can degrade.

RELATED DOCUMENTATION

Network Address Translation Overview on ACX Series | 120

Network Address Port Translation Overview | 121

Enabling Inline Services Interface on ACX Series | 124

Understanding Service Sets
Service Filters in ACX Series | 39
Network Address Translation Address Overload in ACX Series | 122
Network Address Translation Constraints on ACX | 123
Configuring Address Pools for Network Address Port Translation (NAPT) Overview
Configuring Service Sets to Be Applied to Services Interfaces

Configuring the Address and Domain for Services Interfaces

On the AS or Multiservices PIC, you configure a source address for system log messages by including the `address` statement at the `[edit interfaces` *`interface-name`* `unit` *`logical-unit-number`* `family inet]` hierarchy level:

```
address address {
    ...
}
```

Assign an IP address to the interface by configuring the *address* value. The AS or Multiservices PIC generally supports only IP version 4 (IPv4) addresses configured using the `family inet` statement, but IPsec services support IP version 6 (IPv6) addresses as well, configured using the `family inet6` statement.

NOTE: If you configure the same address on multiple interfaces in the same routing instance, Junos OS uses only the first configuration, the remaining address configurations are ignored and can leave interfaces without an address. Interfaces that do not have an assigned address cannot be used as a donor interface for an unnumbered Ethernet interface.

For example, in the following configuration the address configuration of interface xe-0/0/1.0 is ignored:

```
interfaces {
  xe-0/0/0 {
    unit 0 {
      family inet {
        address 192.168.1.1/24;
      }
    }
  }
  xe-0/0/1 {
    unit 0 {
      family inet {
        address 192.168.1.1/24;
      }
    }
}
```

For more information on configuring the same address on multiple interfaces, see *Configuring the Interface Address*.

For information on other addressing properties you can configure that are not specific to service interfaces, see the Junos OS Network Interfaces Library for Routing Devices.

The `service-domain` statement specifies whether the interface is used within the network or to communicate with remote devices. The software uses this setting to determine which default stateful firewall rules to apply, and to determine the default direction for service rules. To configure the domain, include the `service-domain` statement at the `[edit interfaces interface-name unit logical-unit-number]` hierarchy level:

```
service-domain (inside | outside);
```

If you are configuring the interface in a next-hop service-set definition, the `service-domain` setting must match the configuration for the `inside-service-interface` and `outside-service-interface` statements; for more information, see "Configuring Service Sets to be Applied to Services Interfaces" on page 10 .

RELATED DOCUMENTATION

Configuring Default Timeout Settings for Services Interfaces

Example: Configuring an Aggregated Multiservices Interface (AMS) | **1025**

Enabling Session Offloading for Multiservices DPCs

The Junos OS enables you to configure session offloading for Multiservices DPCs on MX Series routers. This enables Fast Update Filters (FUF) at the PIC level for a multiservices interface (ms-*fpc-pic-port*). To configure session offloading, include the `session-offload` statement at the `[edit chassis fpc` *`slot-number`* `pic` *`number`* `adaptive-services service-package extension-provider]` hierarchy level:

```
[edit chassis fpc slot-number pic number adaptive-services service-package extension-provider]
session-offload;
```

Currently, session offloading is supported only for a maximum of one multiservices interface.

NOTE: When session offloading is enabled for a Multiservices PIC, we recommend that you limit dynamic application awareness features for Intrusion Detection and Prevention (IDP) only for that interface.

RELATED DOCUMENTATION

session-offload

2
PART

Network Address Translation

NAT Overview | 53

Stateful NAT64 | 126

Static Source NAT | 131

Static Destination NAT | 156

Network Address Port Translation | 163

Deterministic NAT | 191

NAT Protocol Translation | 202

IPv4 Connectivity Across IPv6-Only Network Using 464XLAT | 238

Port Control Protocol | 243

Secured Port Block Allocation | 264

Port Forwarding | 276

Dynamic Address-Only Source Translation | 287

Inline NAT | 298

Stateless Source Network Prefix Translation for IPv6 | 321

Monitoring NAT | 347

Packet Translation and GRE Tunneling | 360

CHAPTER 3

NAT Overview

IN THIS CHAPTER

- Network Address Translation Overview | 53
- NAT Configuration Overview | 92
- Network Address Translation Overview on ACX Series | 120

Network Address Translation Overview

IN THIS SECTION

- Junos Address Aware Network Addressing Overview | 53
- Sample IPv6 Transition Scenarios | 62
- Junos OS Carrier-Grade NAT Implementation Overview | 64
- Carrier-Grade NAT Feature Comparison for Junos Address Aware by Type of Interface Card | 64
- ALGs Available for Junos OS Address Aware NAT | 71
- ALGs Available by Default for Junos OS Address Aware NAT on ACX500 Router | 77

Junos Address Aware Network Addressing Overview

IN THIS SECTION

- Benefits of NAT | 54
- NAT Concept and Facilities Overview | 54
- IPv4-to-IPv4 Basic NAT | 55

- Deterministic NAPT | 56
- Static Destination NAT | 57
- Twice NAT | 57
- IPv6 NAT | 57
- Application-Level Gateway (ALG) Support | 57
- NAT-PT with DNS ALG | 58
- Dynamic NAT | 58
- Stateful NAT64 | 59
- 464XLAT | 59
- Dual-Stack Lite | 60
- Junos Address Aware Network Addressing Line Card Support | 61

Junos Address Aware Network Addressing provides Network Address Translation (NAT) functionality for translating IP addresses. This is particularly important because the Internet Assigned Numbers Authority (IANA) allocated the last large block of IPv4 addresses in early 2011.

This topic includes the following sections:

Benefits of NAT

NAT supports a wide range of networking goals, including:

- Concealing a set of host addresses on a private network behind a pool of public addresses to protect the host addresses from direct targeting in network attacks and to avoid IPv4 address exhaustion
- Providing the tools to transition to IPv6 based on business requirements and to ensure uninterrupted subscriber and service growth
- Providing IPv4–IPv6 coexistence

NAT Concept and Facilities Overview

Junos Address Aware Network Addressing provides carrier-grade NAT (CGN) for IPv4 and IPv6 networks, and facilitates the transit of traffic between different types of networks.

Junos Address Aware Network Addressing supports a diverse set of NAT translation options:

- Static-source translation—Allows you to hide a private network. It features a one-to-one mapping between the original address and the translated address; the mapping is configured statically. For more information, see "Basic NAT" on page 56 .

- Deterministic NAPT—Eliminates the need for address translation logging by ensuring that the original source IPv4 or IPv6 address and port always map to the same post-NAT IPv4 address and port range.

- Dynamic-source translation— Includes two options: dynamic address-only source translation and *Network Address Port Translation* (NAPT):

  - Dynamic address-only source translation— mdash;A NAT address is picked up dynamically from a source NAT pool and the mapping from the original source address to the translated address is maintained as long as there is at least one active flow that uses this mapping. For more information, see "Dynamic NAT" on page 58 .

  - NAPT—Both the original source address and the source port are translated. The translated address and port are picked up from the corresponding NAT pool. For more information, see "NAPT" on page 56 .

- Static destination translation—Allows you to make selected private servers accessible. It features a one-to-one mapping between the translated address and the destination address; the mapping is configured statically. For more information, see "Static Destination NAT" on page 57 .

- Protocol translation—Allows you to assign addresses from a pool on a static or dynamic basis as sessions are initiated across IPv4 or IPv6 boundaries. For more information, see "Configuring NAT-PT" on page 202 , "NAT-PT with DNS ALG" on page 58 , and "Stateful NAT64" on page 59 .

- Encapsulation of IPv4 packets into IPv6 packets using softwires—Enables packets to travel over softwires to a carrier-grade NAT endpoint where they undergo source-NAT processing to hide the original source address. For more information, see "Tunneling Services for IPv4-to-IPv6 Transition Overview" on page 394 .

Junos Address Aware Network Addressing supports NAT functionality described in IETF RFCs and Internet drafts, as shown in “ Supported NAT and SIP Standards” in Standards Reference.

NOTE: Not all types of NAT are supported on all interface types. See "Carrier-Grade NAT Feature Comparison for Junos Address Aware by Type of Interface Card" on page 64 , which lists features available on supported interfaces.

IPv4-to-IPv4 Basic NAT

Basic Network Address Translation or Basic NAT is a method by which IP addresses are mapped from one group to another, transparent to end users. Network Address Port Translation or NAPT is a method by which many network addresses and their TCP/UDP ports are translated into a single network address and its TCP/UDP ports. Together, these two operations, referred to as traditional NAT, provide a mechanism to connect a realm with private addresses to an external realm with globally unique registered addresses.

Traditional NAT, specified in RFC 3022, *Traditional IP Network Address Translator*, is fully supported by Junos Address Aware Network Addressing. In addition, NAPT is supported for source addresses.

Basic NAT

With Basic NAT, a block of external addresses is set aside for translating addresses of hosts in a private domain as they originate sessions to the external domain. For packets outbound from the private network, Basic NAT translates source IP addresses and related fields such as IP, TCP, UDP, and ICMP header checksums. For inbound packets, Basic NAT translates the destination IP address and the checksums listed above.

Hairpinning is supported for basic NAT.

NAPT

Use NAPT to enable the components of the private network to share a single external address. NAPT translates the transport identifier (for example, TCP port number, UDP port number, or ICMP query ID) of the private network into a single external address. NAPT can be combined with Basic NAT to use a pool of external addresses in conjunction with port translation.

For packets outbound from the private network, NAPT translates the source IP address, source transport identifier (TCP/UDP port or ICMP query ID), and related fields, such as IP, TCP, UDP, and ICMP header checksums. For inbound packets, NAPT translates the destination IP address, the destination transport identifier, and the IP and transport header checksums.

On MX Series routers with MS-MICs and MS-MPCs, if you configure a NAPT44 NAT rule and the source IP address of a spoofed packet is equal to the NAT pool and the NAT rule match condition fails, the packet is continuously looped between the services PIC and the Packet Forwarding Engine. We recommend that you manually clear the session and create a filter to block NAT pool IP spoofing under such conditions.

Hairpinning is supported for NAPT.

Deterministic NAPT

Use deterministic NAPT44 to ensure that the original source IPv4 address and port always map to the same post-NAT IPv4 address and port range, and that the reverse mapping of a given translated external IPv4 address and port are always mapped to the same internal IP address. This eliminates the need for address translation logging. Starting in Junos OS Release 17.4R1, deterministic NAPT64 is supported on the MS-MPC and MS-MIC. Deterministic NAPT64 ensures that the original source IPv6 address and port always map to the same post-NAT IPv4 address and port range, and that the reverse mapping of a given translated external IPv4 address and port are always mapped to the same internal IPv6 address.

Static Destination NAT

Use static destination NAT to translate the destination address for external traffic to an address specified in a destination pool. The destination pool contains one address and no port configuration.

For more information about static destination NAT, see RFC 2663, *IP Network Address Translator (NAT) Terminology and Considerations.*

Twice NAT

In Twice NAT, both the source and destination addresses are subject to translation as packets traverse the NAT router. The source information to be translated can be either address only or address and port. For example, you would use Twice NAT when you are connecting two networks in which all or some addresses in one network overlap with addresses in another network (whether the network is private or public). In traditional NAT, only one of the addresses is translated.

To configure Twice NAT, you must specify both a destination address and a source address for the match direction, pool or prefix, and translation type.

You can configure application-level gateways (ALGs) for ICMP and traceroute under stateful firewall, NAT, or class-of-service (CoS) rules when Twice NAT is configured in the same service set. These ALGs cannot be applied to flows created by the Packet Gateway Control Protocol (PGCP). Twice NAT does not support other ALGs. By default, the Twice NAT feature can affect IP, TCP, and UDP headers embedded in the payload of ICMP error messages.

Twice NAT, specified in RFC 2663, *IP Network Address Translator (NAT) Terminology and Considerations*, is fully supported by Junos Address Aware Network Addressing.

IPv6 NAT

IPv6-to-IPv6 NAT (NAT66), defined in Internet draft draft-mrw-behave-nat66-01, *IPv6-to-IPv6 Network Address Translation (NAT66)*, is fully supported by Junos Address Aware Network Addressing.

Application-Level Gateway (ALG) Support

Junos Address Aware Network Addressing supports a number of ALGs. You can use NAT rules to filter incoming traffic based on ALGS. For more information, see "Network Address Translation Rules Overview" on page 97 .

NAT-PT with DNS ALG

NAT-PT and Domain Name System (DNS) ALG are used to facilitate communication between IPv6 hosts and IPv4 hosts. Using a pool of IPv4 addresses, NAT-PT assigns addresses from that pool to IPv6 nodes on a dynamic basis as sessions are initiated across IPv4 or IPv6 boundaries. Inbound and outbound sessions must traverse the same NAT-PT router so that it can track those sessions. RFC 2766, *Network Address Translation - Protocol Translation (NAT-PT)*, recommends the use of NAT-PT for translation between IPv6-only nodes and IPv4-only nodes, and *not* for IPv6-to-IPv6 translation between IPv6 nodes or IPv4-to-IPv4 translation between IPv4 nodes.

DNS is a distributed hierarchical naming system for computers, services, or any resource connected to the Internet or a private network. The DNS ALG is an application-specific agent that allows an IPv6 node to communicate with an IPv4 node and vice versa.

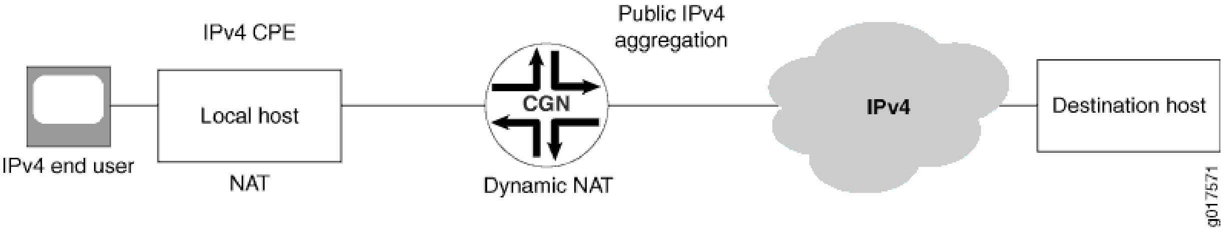
When DNS ALG is employed with NAT-PT, the DNS ALG translates IPv6 addresses in DNS queries and responses to the corresponding IPv4 addresses and vice versa. IPv4 name-to-address mappings are held in the DNS with “A” queries. IPv6 name-to-address mappings are held in the DNS with “AAAA” queries.

NOTE: For IPv6 DNS queries, use the `do-not-translate-AAAA-query-to-A-query` statement at the `[edit applications application *application-name*]` hierarchy level.

Dynamic NAT

Dynamic NAT flow is shown in Figure 2 on page 58 .

Figure 2: Dynamic NAT Flow



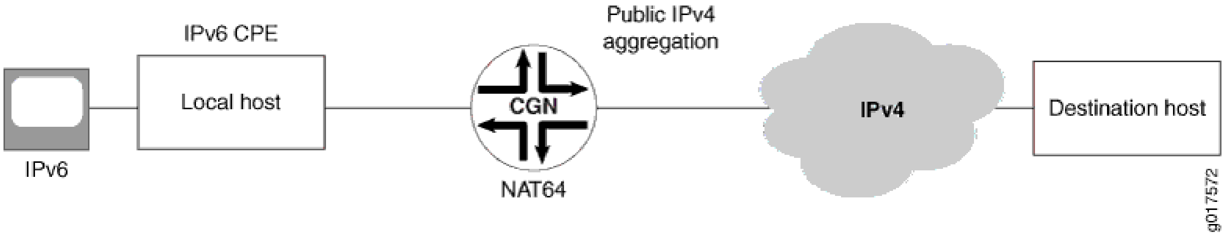
With dynamic NAT, you can map a private IP address (source) to a public IP address drawing from a pool of registered (public) IP addresses. NAT addresses from the pool are assigned dynamically. Assigning addresses dynamically also allows a few public IP addresses to be used by several private hosts, in contrast with an equal-sized pool required by source static NAT.

For more information about dynamic address translation, see RFC 2663, *IP Network Address Translator (NAT) Terminology and Considerations*.

Stateful NAT64

Stateful NAT64 flow is shown in Figure 3 on page 59 .

Figure 3: Stateful NAT64 Flow



Stateful NAT64 is a mechanism to move to an IPv6 network and at the same time deal with IPv4 address depletion. By allowing IPv6-only clients to contact IPv4 servers using unicast UDP, TCP, or ICMP, several IPv6-only clients can share the same public IPv4 server address. To allow sharing of the IPv4 server address, NAT64 translates incoming IPv6 packets into IPv4 (and vice versa).

When stateful NAT64 is used in conjunction with DNS64, no changes are usually required in the IPv6 client or the IPv4 server. DNS64 is out of scope of this document because it is normally implemented as an enhancement to currently deployed DNS servers.

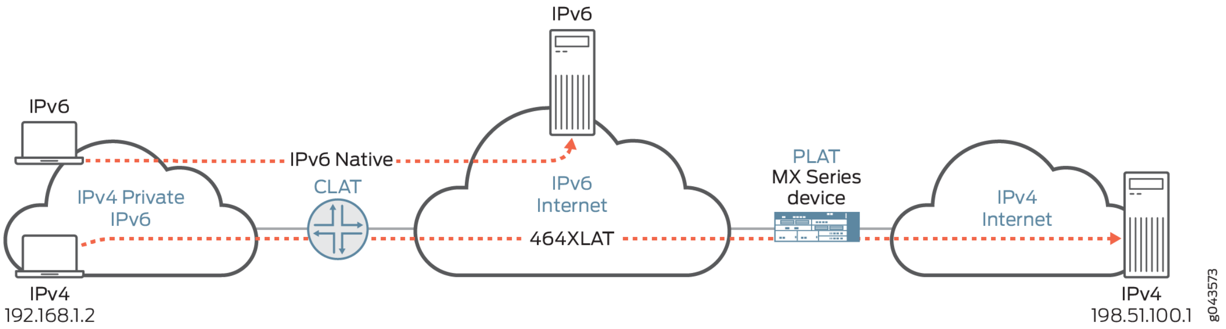
Stateful NAT64, specified in RFC 6146, *Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers*, is fully supported by Junos Address Aware Network Addressing.

464XLAT

Starting in Junos OS Release 17.1R1, you can configure a 464XLAT Provider-Side Translater (PLAT). This is supported only on MS-MICs and MS-MPCs. 464XLAT provides a simple and scalable technique for an IPv4 client with a private address to connect to an IPv4 host over an IPv6 network. 464XLAT only suports IPv4 in the client-server model, so it does not support IPv4 peer-to-peer communication or inbound IPv4 connections.

A customer-side translator (CLAT), which is not a Juniper Networks product, translates the IPv4 packet to IPv6 by embedding the IPv4 source and destination addresses in IPv6 /96 prefixes, and sends the packet over an IPv6 network to the PLAT. The PLAT translates the packet to IPv4, and sends the packet to the IPv4 host over an IPv4 network (see Figure 4 on page 60).

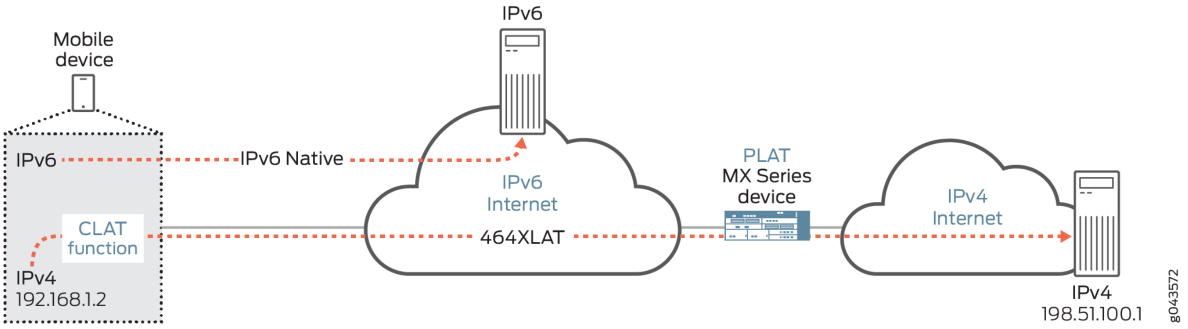
Figure 4: 464XLAT Wireline Flow



XLAT464 provides the advantages of not having to maintain an IPv4 network and not having to assign additional public IPv4 addresses.

The CLAT can reside on the end user mobile device in an IPv6-only mobile network, allowing mobile network providers to roll out IPv6 for their users *and* support IPv4-only applications on mobile devices (see Figure 5 on page 60).

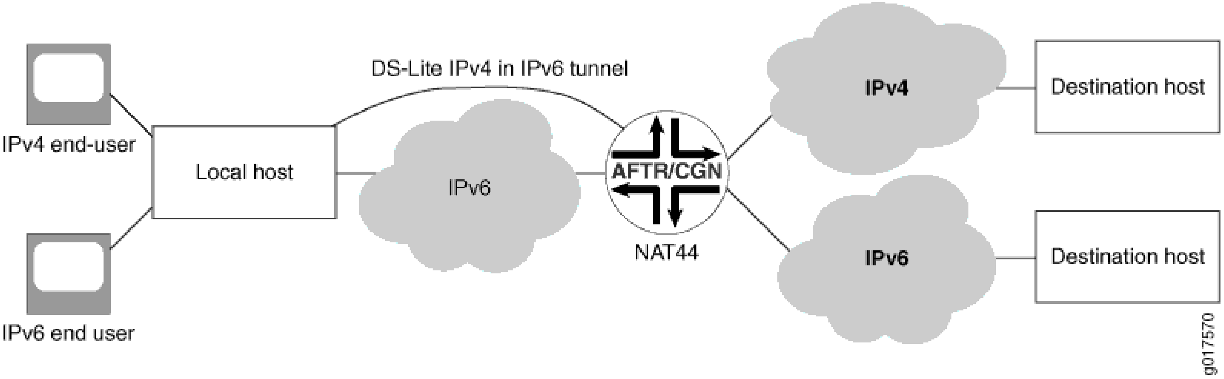
Figure 5: 464XLAT Wireless Flow



Dual-Stack Lite

Dual-stack lite (DS-Lite) flow is shown in Figure 6 on page 61 .

Figure 6: DS-Lite Flow



DS-Lite employs IPv4-over-IPv6 tunnels to cross an IPv6 access network to reach a carrier-grade IPv4-IPv4 NAT. This facilitates the phased introduction of IPv6 on the Internet by providing backward compatibility with IPv4.

DS-Lite is supported on MX series routers with MS-DPCs and on M Series routers with MS-100, MS-400, and MS-500 MultiServices PICS. Starting in Junos OS release 17.4R1, DS-Lite is supported on MX Series routers with MS-MPCs and MS-MICs.Starting in Junos OS release 19.2R1, DS-Lite is supported on MX Virtual Chassis and MX Broadband Network Gateway (BNG) routers.

Junos Address Aware Network Addressing Line Card Support

Junos Address Aware Network Addressing technologies are available on the following line cards:

- MultiServices Dense Port Concentrator (MS-DPC)
- MS-100, MS-400, and MS-500 MultiServices PICS
- MultiServices Modular Port Concentrator (MS-MPC) and MultiServices Modular Interface Card (MS-MIC)
- Modular Port Concentrators (inline NAT).

For a listing of the specific NAT types supported on each type of card, see "Carrier-Grade NAT Feature Comparison for Junos Address Aware by Type of Interface Card" on page 64 .

SEE ALSO

ALGs Available for Junos OS Address Aware NAT | 71

Sample IPv6 Transition Scenarios

IN THIS SECTION

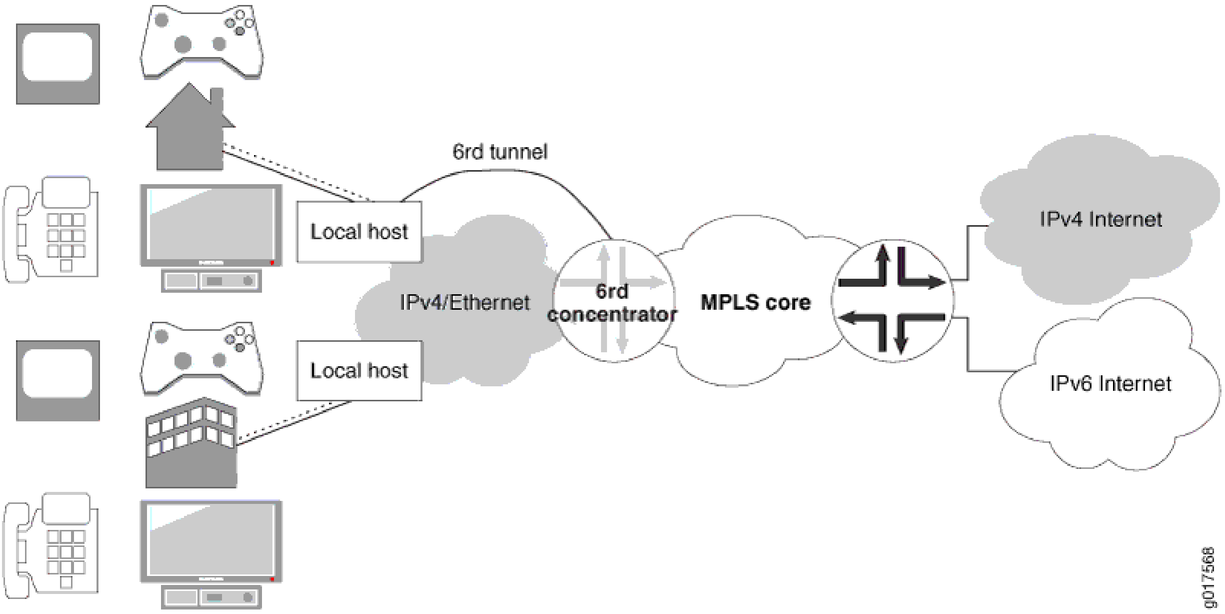
- Example 1: IPv4 Depletion with a Non-IPv6 Access Network | 62
- Example 2: IPv4 Depletion with an IPv6 Access Network | 63
- Example 3: IPv4 Depletion for Mobile Networks | 63

The Junos OS supports many IPv6 transition scenarios required by Junos OS customers. The following are selected examples:

Example 1: IPv4 Depletion with a Non-IPv6 Access Network

Figure 7 on page 62 depicts a scenario in which the Internet service provider (ISP) has not significantly changed its IPv4 network. This approach enables IPv4 hosts to access the IPv4 Internet and IPv6 hosts to access the IPv6 Internet. A dual-stack host can be treated as an IPv4 host when it uses the IPv4 access service, and as an IPv6 host when it uses the IPv6 access service.

Figure 7: IPv4 Depletion Solution - IPv4 Access Network

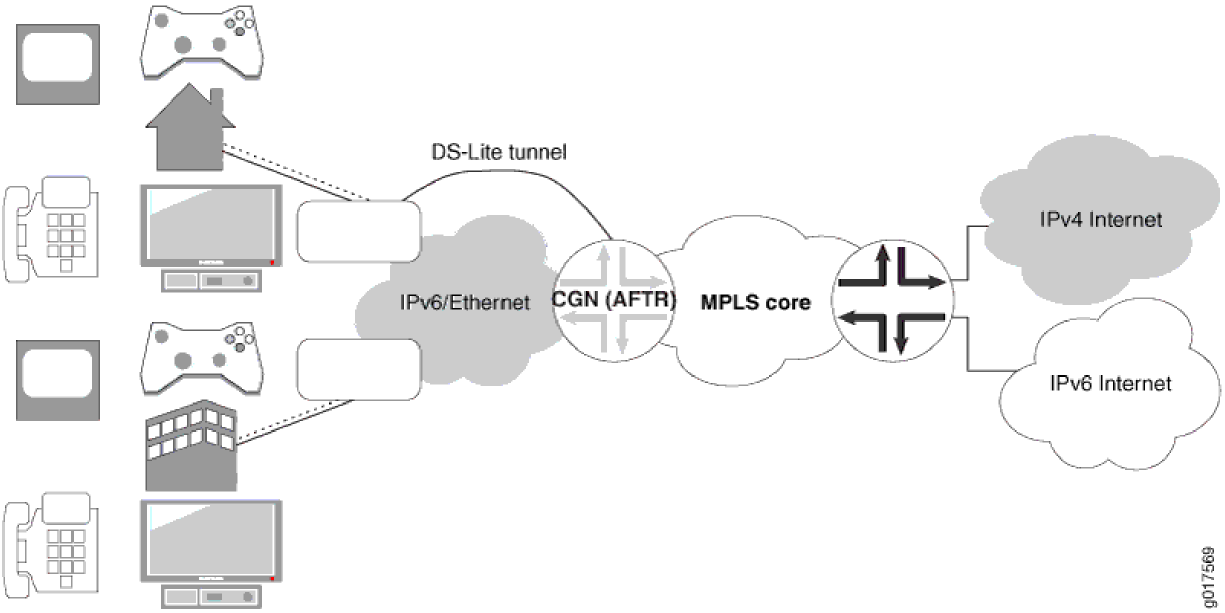


Two new types of devices must be deployed in this approach: a dual-stack home gateway and a dual-stack carrier-grade Network Address Translation (NAT). The dual-stack home gateway integrates IPv4 forwarding and v6-over-v4 tunneling functions. It can also integrate a v4-v4 NAT function. The dual-stack carrier-grade NAT (CGN) integrates v6-over-v4 tunneling and carrier-grade v4-v4 NAT functions.

Example 2: IPv4 Depletion with an IPv6 Access Network

In the scenario shown in Figure 8 on page 63 , the ISP network is IPv6-only.

Figure 8: IPv4 Depletion Solution - IPv6 Access Network



The dual-stack lite (DS-Lite) solution accommodates IPv6-only ISPs. The best business model for this approach is that the customer premises equipment (CPE) has integrated the functions for tunneling IPv6 to an IPv4 backbone, tunneling IPv4 to an IPv6 backbone, and can automatically detect which solution is required.

Not all customers of a given ISP must switch from IPv4 access to IPv6 access simultaneously; in fact, transition can be managed better by switching groups of customers (for example, all those connected to a single point of presence) on an incremental basis. Such an incremental approach should prove easier to plan, schedule, and execute than an across-the-board conversion.

Example 3: IPv4 Depletion for Mobile Networks

The complexity of mobile networks necessitates a flexible migration approach to ensure minimal disruption and maximum backward compatibility during transition. NAT64 can be used to enable IPv6 devices to communicate to IPv4 hosts without modifying the clients.

Junos OS Carrier-Grade NAT Implementation Overview

Junos OS enables you to implement and scale a Carrier-Grade Network Address Translation (CGNAT) solution based on the type of services interfaces used for your implementation:

- MultiServices Denser Port Concentrator (MS-DPC)—The layer 3 services package is used to configure NAT for MS-DPC adaptive services PICs. This solution provides the NAT functionality described in "Junos Address Aware Network Addressing Overview" on page 53 .

- MS-100, MS-400, and MS-500 MultiServices PICS—The layer 3 services package is used to configure NAT for multiservices PICs. This solution provides the NAT functionality described in "Junos Address Aware Network Addressing Overview" on page 53 .

- MultiServices Modular Port Concentrator (MS-MPC) and MultiServices Modular Interface Card (MS-MIC)—MS-MPCs and MS-MICs are pre-configured to enable configuration of carrier-grade NAT. This solution provides the NAT functionality described in "Junos Address Aware Network Addressing Overview" on page 53 .

- Inline NAT for Modular Port Concentrator (MPC) Line Cards)—Inline NAT leverages the services capabilities of MPC line cards, allowing a cost-effective implementation of NAT functionality on the data plane, as described in "Inline Network Address Translation Overview" on page 298 .

SEE ALSO

Carrier-Grade NAT Implementation: Best Practices | **106**

Example: Configuring Basic NAT44 | **147**

Carrier-Grade NAT Feature Comparison for Junos Address Aware by Type of Interface Card

Table 6 on page 65 summarizes feature differences among the Junos OS carrier-grade NAT implementations.

Starting in Junos OS release 17.2R1, inline NAT is supported on the MPC5E and MPC6E.

Starting in Junos OS release 17.4R1, inline NAT is supported on the MPC7E, MPC8E, and MPC9E.

Table 6: Carrier-Grade NAT—Feature Comparison by Platform

Feature	MS-DPC MS-100 MS-400 MS-500	MS-MPC MS-MIC	MPC1, MPC2, MPC3, MPC5E, MPC6E, MPC7E, MPC8E, and MPC9E *Inline NAT*
Static Source NAT	yes	yes	yes
Dynamic Source NAT - Address Only	yes	yes	no
Dynamic Source NAT - NAPT Port Translation with Secured Port Block Allocation	yes	yes (Dynamic Source NAT - NAPT Port Translation with Secured Port Block Allocation supported for MS-MPC and MS-MIC starting in Junos OS Release 14.2R2)	no
Dynamic Source NAT - NAPT44 Port Translation with Deterministic Port Block Allocation	yes	yes (Dynamic Source NAT - NAPT44 Port Translation with Deterministic Port Block Allocation supported for MS-MPC and MS-MIC starting in Junos OS release 17.3R1, in Junos OS release 14.2R7 and later 14.2 releases, in 15.1R3 and later 15.1 releases, and in 16.1R5 and later 16.1 releases)	no

Table 6: Carrier-Grade NAT—Feature Comparison by Platform *(Continued)*

Feature	MS-DPC MS-100 MS-400 MS-500	MS-MPC MS-MIC	MPC1, MPC2, MPC3, MPC5E, MPC6E, MPC7E, MPC8E, and MPC9E *Inline NAT*
Dynamic Source NAT - NAPT64 Port Translation with Deterministic Port Block Allocation	No	yes (Dynamic Source NAT - NAPT64 Port Translation with Deterministic Port Block Allocation supported for MS-MPC and MS-MIC starting in Junos OS release 17.4R1)	No
Static Destination NAT	yes	yes	yes **NOTE**: Destination NAT can be implemented indirectly. See "Inline Network Address Translation Overview" on page 298
Twice NAT	yes	yes (Twice NAT supported for MS-MPC and MS-MIC starting in Junos OS Release 15.1R1)	yes **NOTE**: Twice NAT can be implemented indirectly. See "Inline Network Address Translation Overview" on page 298
NAPT - Preserve Parity and Range	yes	yes (NAPT - Preserve Parity and Range supported for MS-MPC and MS-MIC starting in Junos OS release 15.1R1)	no
NAPT - APP/EIF/EIM	yes	yes	no

Table 6: Carrier-Grade NAT—Feature Comparison by Platform *(Continued)*

Feature	MS-DPC MS-100 MS-400 MS-500	MS-MPC MS-MIC	MPC1, MPC2, MPC3, MPC5E, MPC6E, MPC7E, MPC8E, and MPC9E *Inline NAT*
IKE ALG	no	yes (Starting in Junos OS Release 14.2R7, 15.1R5, 16.1R2, and 17.1R1)	no
Stateful NAT64	yes	yes	no
Stateful NAT64 with APP/EIM/EIF	no	yes	no
Stateful NAT64 with ALGs • FTP • IKE • TFTP • SIP • RTSP • PPPT	no	yes	no
DS-Lite	yes	yes (DS-Lite supported for MS-MPC and MS-MIC starting in Junos OS release 17.4R1)	no
6rd	yes	no	no
6to4	yes	no	no

Table 6: Carrier-Grade NAT—Feature Comparison by Platform *(Continued)*

Feature	MS-DPC MS-100 MS-400 MS-500	MS-MPC MS-MIC	MPC1, MPC2, MPC3, MPC5E, MPC6E, MPC7E, MPC8E, and MPC9E *Inline NAT*
464XLAT	no	yes (starting in Junos OS Release 17.1R1)	no
Overlap Address Across NAT Pool	yes	yes	no
Overload Pool	yes	no	no
Port Control Protocol	yes	yes (Port Control Protocol with NAPT44 is supported for MS-MPC and MS-MIC starting in Junos OS Release 17.4R1. Starting in Junos OS Release 18.2R1, Port Control Protocol on the MS-MPC and MS-MIC supports DS-Lite. PCP provides a mechanism to control the forwarding of incoming packets by upstream devices such as NAT44 and firewall devices, and a mechanism to reduce application keepalive traffic).	no
CGN-PIC	yes	no	no
AMS Support	no	yes	no

Table 6: Carrier-Grade NAT—Feature Comparison by Platform *(Continued)*

Feature	MS-DPC MS-100 MS-400 MS-500	MS-MPC MS-MIC	MPC1, MPC2, MPC3, MPC5E, MPC6E, MPC7E, MPC8E, and MPC9E *Inline NAT*
Port forwarding	yes	yes (Port forwarding is supported for MS-MPC and MS-MIC starting in Junos OS Release 17.4R1.)	no
No translation	yes	yes (No translation supported for MS-MPC and MS-MIC starting in Junos OS Release 15.1R1)	yes

Table 7 on page 69 summarizes availability of translation types by type of line card.

Table 7: Carrier-Grade NAT Translation Types

Translation Type	MS-DPC MS-100 MS-400 MS-500	MS-MPC MS-MIC	MPC1, MPC2, MPC3, MPC5E, MPC6E, MPC7E, MPC8E, and MPC9E *Inline NAT*
`basic-nat44`	yes	yes	yes
`basic-nat66`	yes	no	no
`basic-nat-pt`	yes	no	no

Table 7: Carrier-Grade NAT Translation Types *(Continued)*

Translation Type	MS-DPC MS-100 MS-400 MS-500	MS-MPC MS-MIC	MPC1, MPC2, MPC3, MPC5E, MPC6E, MPC7E, MPC8E, and MPC9E *Inline NAT*
`deterministic-napt44`	yes	yes (`deterministic-napt44` supported for MS-MPC and MS-MIC starting in Junos OS release 17.3R1, in Junos OS release 14.2R7 and later 14.2 releases, in 15.1R3 and later 15.1 releases, and in 16.1R5 and later 16.1 releases)	no
`deterministic-napt64`	no	yes (`deterministic-napt64` supported for MS-MPC and MS-MIC starting in Junos OS release 17.4R1)	no
`dnat-44`	yes	yes	no
`dynamic-nat44`	yes	yes	no
`napt-44`	yes	yes	no
`napt-66`	yes	no	no
`napt-pt`	yes	no	no
`stateful-nat464`	no	yes (starting in Junos OS Release 17.1R1)	no
`stateful-nat64`	yes	yes	no

Table 7: Carrier-Grade NAT Translation Types *(Continued)*

Translation Type	MS-DPC MS-100 MS-400 MS-500	MS-MPC MS-MIC	MPC1, MPC2, MPC3, MPC5E, MPC6E, MPC7E, MPC8E, and MPC9E *Inline NAT*
`twice-basic-nat-44`	yes	yes (`twice-dynamic-nat-44` supported for MS-MPC and MS-MIC starting in Junos OS Release 15.1R1)	yes (`twice-basic-nat-44` supported for inline NAT starting in Junos OS Release 15.1R1)
`twice-dynamic-nat-44`	yes	yes (`twice-dynamic-nat-44` supported for MS-MPC and MS-MIC starting in Junos OS Release 15.1R1)	no
`twice-dynamic-napt-44`	yes	yes (`twice-dynamic-napt-44` supported for MS-MPC and MS-MIC starting in Junos OS Release 15.1R1)	no

ALGs Available for Junos OS Address Aware NAT

The following Application Level Gateways (ALGs) listed in Table 8 on page 72 are supported for NAT processing on the listed platforms.

To view the implementation details (port, protocol, and so on) for these Junos OS default applications, locate the Junos OS Default ALG Name in the table and then look up the listed name in the `groups`. For example, for details about TFTP, look up `junos-tftp` as shown.

TIP: The Junos OS provides the `junos-alg`, which enables other ALGs to function by handling ALG registrations, causing slow path packets to flow through registered ALGs, and transferring ALG events to the ALG plug-ins. The `junos-alg` ALG is automatically available on the MS-MPC and MS-MIC platforms and does not require further configuration.

NOTE: The remote shell (RSH) and remote login (rlogin) application layer gateways (ALGs) are not supported with network address port translation (NAPT) on MX Series routers with MS-MICs and MS-MPCs.

```
user@host# show groups junos-defaults applications application junos-tftp
application-protocol tftp;
protocol udp;
destination-port 69;
```

Table 8 on page 72 summarizes the ALGs available for Junos OS Address Aware NAT for services interfaces cards.

Table 8: ALGs Available for NAT by Type of Interface Card

ALG	MS-DPC	MS-MPC, MS-MIC	Junos OS Default ALG Name
Basic TCP ALG	yes	yes	**NOTE**: Specific Junos OS ALGs are not supported. However, a feature called TCP tracker, available by default, performs segment ordering and retransmit and connection tracking, validations for TCP connections.
Basic UDP ALG	yes	yes	**NOTE**: TCP tracker performs limited integrity and validation checks for UDP.
BOOTP	yes	no	• junos-bootpc • junos-bootps

Table 8: ALGs Available for NAT by Type of Interface Card *(Continued)*

ALG	MS-DPC	MS-MPC, MS-MIC	Junos OS Default ALG Name
DCE RPC Services	yes	yes	• `junos-dce-rpc-portmap` • `junos-dcerpc-endpoint-mapper-service` • `junos-dcerpc-msexchange-directory-nsp` • `junos-dcerpc-msexchange-directory-rfr` • `junos-dcerpc-msexchange-information-store`
DNS	yes	yes	• `junos-dns-udp`
DNS	no	no	• `junos-dns-tcp`
FTP	yes	yes	• `junos-ftp`
Gatekeeper RAS (Starting in Junos OS Release 17.1R1)	no	yes	• `junos-h323-ras`
H323	no	yes	• `junos-h323`

Table 8: ALGs Available for NAT by Type of Interface Card *(Continued)*

ALG	MS-DPC	MS-MPC, MS-MIC	Junos OS Default ALG Name
ICMP	yes	yes **NOTE**: In Junos OS Release 14.1 and earlier, ICMP messages are handled by default, but PING ALG support is not provided. Starting In Junos OS 14.2, ICMP messages are handled by default and PING ALG support is provided.	• `junos-icmp-all` • `junos-icmp-ping`
IIOP	yes	no	• `junos-iiop-java` • `junos-iiop-orbix`
IKE ALG	no	yes **NOTE**: Starting in Junos OS Release 14.2R7, 15.1R5, 16.1R2, and 17.1R1, the IKE ALG ALG is supported on MS-MPCs and MS-MICs.	• `junos-ike`
IP	yes	The TCP tracker, available by default on these platforms, performs limited integrity and validation checks.	• `junos-ip`
NETBIOS	yes	no	• `junos-netbios-datagram` • `junos-netbios-name-tcp` • `junos-netbios-name-udp` • `junos-netbios-session`
NETSHOW	yes	no	• `junos-netshow`

Table 8: ALGs Available for NAT by Type of Interface Card *(Continued)*

ALG	MS-DPC	MS-MPC, MS-MIC	Junos OS Default ALG Name
PPTP	yes	yes	• `junos-pptp`
REALAUDIO	yes	no	• `junos-realaudio`
Sun RPC and RPC Port Map Services	yes	yes	• `junos-rpc-portmap-tcp` • `junos-rpc-portmap-udp`
RTSP	yes	yes	• `junos-rtsp`
SIP	yes	Yes	• `junos-sip` The SIP `callid` is *not* translated in `register` messages. **NOTE**: SIP sessions are limited to 12 hours (720 minutes) for NAT processing on the MS-MIC and MS-MPC interface cards. SIP sessions on the MS-DPC have no time limits.
SNMP	yes	No	• `junos-snmp-get` • `junos-snmp-get-next` • `junos-snmp-response` • `junos-snmp-trap`
SQLNET	yes	yes	• `junos-sqlnet`

Table 8: ALGs Available for NAT by Type of Interface Card *(Continued)*

ALG	MS-DPC	MS-MPC, MS-MIC	Junos OS Default ALG Name
TFTP	yes	yes	• `junos-tftp`
Traceroute	yes	yes	• `junos-traceroute`
Unix Remote Shell Service	yes	yes **NOTE**: Remote Shell (RSH) ALG is not supported for network address port translation (NAPT).	• `junos-rsh`
WINFrame	yes	No	• `junos-citrix-winframe` • `junos-citrix-winframe-udp`
TALK-UDP	No	Yes	• `junos-talk-udp`
MS RPC	No	Yes	• `junos-rpc-portmap-tcp` • `junos-rpc-portmap-udp` • `junos-rpc-services-tcp` • `junos-rpc-services-udp`

SEE ALSO

ALG Descriptions | 475

ALGs Available by Default for Junos OS Address Aware NAT on ACX500 Router

IN THIS SECTION

- ALG Support Details | 78

The following Application Level Gateways (ALGs) listed in Table 9 on page 77 are supported for NAT processing on ACX500 routers.

To view the implementation details (port, protocol, and so on) for these Junos OS default applications, locate the Junos OS Default ALG Name in the table and then look up the listed name in the `groups`. For example, for details about TFTP, look up `junos-tftp` as shown.

NOTE: The ALG for NAT is supported only on the ACX500 indoor routers.

TIP: The Junos OS provides the `junos-alg`, which enables other ALGs to function by handling ALG registrations, causing slow path packets to flow through registered ALGs, and transferring ALG events to the ALG plug-ins. The `junos-alg` ALG is automatically available on the ACX500 router and does not require further configuration.

NOTE: The remote login (rlogin) application layer gateways (ALGs) are not supported with network address port translation (NAPT) on ACX500 router.

Table 9: ALGs Available by Default

ALG	ACX500 Router	Junos OS Default ALG Name
Basic TCP ALG	yes	**NOTE**: Specific Junos OS ALGs are not supported. However, a feature called TCP tracker, available by default, performs segment ordering and retransmit and connection tracking, validations for TCP connections.

Table 9: ALGs Available by Default *(Continued)*

ALG	ACX500 Router	Junos OS Default ALG Name
Basic UDP ALG	yes	**NOTE**: TCP tracker performs limited integrity and validation checks for UDP.
DNS	yes	• `junos-dns-tcp` • `junos-dns-udp`
FTP	yes	• `junos-ftp`
ICMP	yes **NOTE**: ICMP messages are handled by default, but PING ALG support is not provided.	• `junos-icmp-all`
TFTP	yes	• `junos-tftp`
Unix Remote Shell Service	yes **NOTE**: Remote Shell (RSH) ALG is not supported for network address port translation (NAPT).	• `junos-rsh`

ALG Support Details

This section includes details about the ALGs. It includes the following:

Basic TCP

This ALG performs basic sanity checking on TCP packets. If it finds errors, it generates the following anomaly events and system log messages:

- TCP source or destination port zero
- TCP header length check failed

- TCP sequence number zero and no flags are set
- TCP sequence number zero and FIN/PSH/RST flags are set
- TCP FIN/RST or SYN(URG|FIN|RST) flags are set

The TCP ALG performs the following steps:

1. When the router receives a SYN packet, the ALG creates TCP forward and reverse flows and groups them in a *conversation*. It tracks the TCP three-way handshake.
2. The SYN-defense mechanism tracks the TCP connection establishment state. It expects the TCP session to be established within a small time interval (currently 4 seconds). If the TCP three-way handshake is not established in that period, the session is terminated.
3. A keepalive mechanism detects TCP sessions with nonresponsive endpoints.
4. ICMP errors are allowed only when a flow matches the selector information specified in the ICMP data.

Basic UDP

This ALG performs basic sanity checking on UDP headers. If it finds errors, it generates the following anomaly events and system log messages:

- UDP source or destination port 0
- UDP header length check failed

The UDP ALG performs the following steps:

1. When it receives the first packet, the ALG creates bidirectional flows to accept forward and reverse UDP session traffic.
2. If the session is idle for more than the maximum allowed idle time (the default is 30 seconds), the flows are deleted.
3. ICMP errors are allowed only when a flow matches the selector information specified in the ICMP data.

DNS

The Domain Name System (DNS) ALG handles data associated with locating and translating domain names into IP addresses. The ALG typically runs on port 53. The ALG monitors DNS query and reply packets and supports only UDP traffic. The ALG does not support payload translations. The DNS ALG closes the session only when a reply is received or an idle timeout is reached.

The following is an example for configuring DNS ALG:

1. Creating NAT interface.

```
[edit]
services {
    service-set set-dns {
        nat-rules nat-dns;
        interface-service {
            service-interface ms-0/2/0;
        }
    }
```

2. Configuring NAT pool.

```
[edit]
services {
    nat {
        pool p-napt {
            address 10.1.1.1/32;
        }
    }
}
```

3. Defining NAT rules for DNS ALG.

```
[edit]
services {
    nat {
        rule nat-dns {
            match-direction input;
            term term1 {
                from {
                    source-address {
                        10.50.50.2/32;
                    }
                    applications junos-dns-udp;;
                }
                then {
                    translated {
                        source-pool p-napt;
                            translation-type {
                            basic-nat44;
                        }
                    }
                }
        }
    }
}
```

4. Binding service sets to the interface.

```
[edit]
interfaces {
    ge-0/1/0 {
        media-type copper;
        unit 0 {
            family inet {
                service {
                    input {
                    service-set set-dns;
                    }
                    output {
                    service-set set-dns;
                }
            }
            address 10.50.50.1/24;
        }
    }
    }
    ge-0/1/1 {
        media-type copper;
        unit 0 {
            family inet {
            address 10.60.60.1/24;
        }
    }
    }
    ms-0/2/0 {
        unit 0 {
            family inet;
        }
    }
}
```

FTP

FTP is the File Transfer Protocol, specified in RFC 959. In addition to the main control connection, data connections are also made for any data transfer between the client and the server; and the host, port, and direction are negotiated through the control channel.

For non-passive-mode FTP, Junos OS stateful firewall service scans the client-to-server application data for the PORT command, which provides the IP address and port number to which the server connects. For passive-mode FTP, Junos OS stateful firewall service scans the client-to-server application data for the PASV command and then scans the server-to-client responses for the 227 response, which contains the IP address and port number to which the client connects.

There is an additional complication: FTP represents these addresses and port numbers in ASCII. As a result, when addresses and ports are rewritten, the TCP sequence number might be changed, and thereafter the NAT service needs to maintain this delta in SEQ and ACK numbers by performing sequence NAT on all subsequent packets.

Support for stateful firewall and NAT services requires that you configure the FTP ALG on TCP port 21 to enable the FTP control protocol. The ALG performs the following tasks:

- Automatically allocates data ports and firewall permissions for dynamic data connection
- Creates flows for the dynamically negotiated data connection
- Monitors the control connection in both active and passive modes
- Rewrites the control packets with the appropriate NAT address and port information

On ACX500, for passive FTP to work properly without FTP application layer gateway (ALG) enabled (by not specifying the `application junos-ftp` statement at the `[edit services nat rule` *`rule-name`* `term` *`term-name`* `from]` hierarchy level), you must enable the address pooling paired (APP) functionality enabled (by including the `address-pooling` statement at the `[edit services nat rule` *`rule-name`* `term` *`term-name`* `then translated]` hierarchy level). Such a configuration causes the data and control FTP sessions to receive the same NAT address.

The following is an example for configuring FTP ALG:

1. Creating NAT interface.

```
[edit]
services {
    service-set set-ftp {
        nat-rules nat-ftp;
        interface-service {
            service-interface ms-0/2/0;
        }
    }
```

2. Configuring NAT pool.

```
[edit]
services {
    nat {
        pool p-napt {
            address 10.30.30.0/24;
            port {
            range low 9000 high 9010;
        }
    }
}
```

3. Defining NAT rules for FTP ALG.

```
[edit]
services {
    nat {
        rule nat-ftp {
            match-direction input;
            term term1 {
                from {
                    source-address {
                        10.10.10.0/24;
                    }
                    applications junos-ftp;
                }
                then {
                    translated {
                        source-pool p-napt;
                            translation-type {
                            napt-44;
                        }
                    }
                }
        }
    }
}
```

4. Binding service sets to the interface.

```
[edit]
interfaces {
    ge-0/1/0 {
        media-type copper;
        unit 0 {
            family inet {
                service {
                    input {
                    service-set set-ftp;
                    }
                    output {
                    service-set set-ftp;
                }
            }
            address 10.10.10.1/24;
        }
    }
    }
    ge-0/1/1 {
        media-type copper;
        unit 0 {
            family inet {
            address 10.10.10.1/24;
        }
    }
    }
    ms-0/2/0 {
        unit 0 {
            family inet;
        }
    }
}
```

ICMP

The Internet Control Message Protocol (ICMP) is defined in RFC 792. The Junos OS allows ICMP messages to be filtered by specific type or specific type code value. ICMP error packets that lack a specifically configured type and code are matched against any existing flow in the opposite direction to check for the legitimacy of the error packet. ICMP error packets that pass the filter matching are subject to NAT translation.

The ICMP ALG always tracks ping traffic statefully using the ICMP sequence number. Each echo reply is forwarded only if there is an echo request with the corresponding sequence number. For any ping flow, only 20 echo requests can be forwarded without receiving an echo reply. When you configure dynamic NAT, the PING packet identifier is translated to allow additional hosts in the NAT pool to use the same identifier.

Support for NAT services requires that you configure the ICMP ALG if the protocol is needed. You can configure the ICMP type and code for additional filtering.

TFTP

The Trivial File Transfer Protocol (TFTP) is specified in RFC 1350. The initial TFTP requests are sent to UDP destination port 69. Additional flows can be created to **get** or **put** individual files. Support of NAT services requires that you configure the TFTP ALG for UDP destination port 69.

The following is an example for configuring TFTP ALG:

1. Creating NAT interface.

```
[edit]
services {
    service-set set-tftp {
        nat-rules nat-tftp;
        interface-service {
            service-interface ms-0/2/0;
        }
    }
```

2. Configuring NAT pool.

```
[edit]
services {
    nat {
        pool p-napt {
            address 10.1.1.1/32;
        }
    }
}
```

3. Defining NAT rules for TFTP ALG.

```
[edit]
services {
    nat {
        rule nat-tftp {
            match-direction input;
            term term1 {
                from {
                    source-address {
                        10.50.50.2/32;
                    }
                    applications junos-tftp;
                }
                then {
                    translated {
                        source-pool p-napt;
                            translation-type {
                            dynamic-nat44;
                        }
                    }
                }
        }
    }
}
```

4. Binding service sets to the interface.

```
[edit]
interfaces {
    ge-0/1/0 {
        media-type copper;
        unit 0 {
            family inet {
                service {
                    input {
```

```
                    service-set set-tftp;
                    }
                    output {
                    service-set set-tftp;
                }
            }
            address 10.50.50.1/24;
        }
    }
    }
    ge-0/1/1 {
        media-type copper;
        unit 0 {
            family inet {
            address 10.60.60.1/24;
        }
    }
    }
    ms-0/2/0 {
        unit 0 {
            family inet;
        }
    }
}
```

UNIX Remote-Shell Services

Three protocols form the basis for UNIX remote-shell services:

- Exec—Remote command execution; enables a user on the client system to execute a command on the remote system. The first command from client (`rcmd`) to server (`rshd`) uses well-known TCP port 512. A second TCP connection can be opened at the request of `rcmd`. The client port number for the second connection is sent to the server as an ASCII string.

- Login—Better known as `rlogin`; uses well-known TCP port 513. For details, see RFC 1282. No special firewall processing is required.

- Shell—Remote command execution; enables a user on the client system to execute a command on the remote system. The first command from client (`rcmd`) to server (`rshd`) uses well-known TCP port 514. A second TCP connection can be opened at the request of `rcmd`. The client port number for the second connection is sent to the server as an ASCII string.

NAT remote-shell services require that any dynamic source port assigned be within the port range 512 to 1023. If you configure a NAT pool, this port range is reserved exclusively for remote shell applications.

The following is an example for configuring RSH ALG:

1. Creating NAT interface.

```
[edit]
services {
    service-set set-rsh {
        nat-rules nat-rsh;
        interface-service {
            service-interface ms-0/2/0;
        }
    }
```

2. Configuring NAT pool.

```
[edit]
services {
    nat {
        pool p-napt {
            address 10.1.1.1/32;
        }
    }
}
```

3. Defining NAT rules for RSH ALG.

```
[edit]
services {
    nat {
        rule nat-rsh {
            match-direction input;
            term term1 {
                from {
                    source-address {
                        510.0.50.2/32;
                    }
                    applications junos-rsh;
                }
                then {
                    translated {
                        source-pool p-napt;
                            translation-type {
                            dynamic-nat44;
                        }
                    }
                }
        }
    }
}
```

4. Binding service sets to the interface.

```
[edit]
interfaces {
    ge-0/1/0 {
        media-type copper;
        unit 0 {
            family inet {
                service {
                    input {
                    service-set set-rsh;
                    }
                    output {
                    service-set set-rsh;
                }
            }
            address 10.50.50.1/24;
        }
    }
    }
    ge-0/1/1 {
        media-type copper;
        unit 0 {
            family inet {
            address 10.60.60.1/24;
        }
    }
    }
    ms-0/2/0 {
        unit 0 {
            family inet;
        }
    }
}
```

SEE ALSO

- Junos Network Secure Overview | 546
- Configuring Stateful Firewall Rules | 550
- Understanding Service Sets
- Configuring Service Sets to Be Applied to Services Interfaces

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
19.2R1	Starting in Junos OS release 19.2R1, DS-Lite is supported on MX Virtual Chassis and MX Broadband Network Gateway (BNG) routers.
17.4R1	Starting in Junos OS Release 17.4R1, deterministic NAPT64 is supported on the MS-MPC and MS-MIC.
17.4R1	Starting in Junos OS release 17.4R1, DS-Lite is supported on MX Series routers with MS-MPCs and MS-MICs.
17.4R1	Starting in Junos OS release 17.4R1, inline NAT is supported on the MPC7E, MPC8E, and MPC9E.
17.4R1	Dynamic Source NAT - NAPT64 Port Translation with Deterministic Port Block Allocation supported for MS-MPC and MS-MIC
17.4R1	DS-Lite supported for MS-MPC and MS-MIC
17.4R1	`deterministic-napt64` supported for MS-MPC and MS-MIC
17.4.R1	Port forwarding is supported for MS-MPC and MS-MIC
17.2R1	Starting in Junos OS release 17.2R1, inline NAT is supported on the MPC5E and MPC6E.

17.1R4	Port Control Protocol with NAPT44 is supported for MS-MPC and MS-MIC
17.1R1	Starting in Junos OS Release 17.1R1, you can configure a 464XLAT Provider-Side Translater (PLAT).
17.1R1	464XLAT
17.1R1	`stateful-nat464`
17.1R1	Gatekeeper RAS (Starting in Junos OS Release 17.1R1)
15.1R1	Twice NAT supported for MS-MPC and MS-MIC
15.1R1	NAPT - Preserve Parity and Range supported for MS-MPC and MS-MIC
15.1R1	No translation supported for MS-MPC and MS-MIC
15.1R1	`twice-dynamic-nat-44` supported for MS-MPC and MS-MIC
15.1R1	`twice-basic-nat-44` supported for inline NAT
15.1R1	`twice-dynamic-nat-44` supported for MS-MPC and MS-MIC
15.1R1	`twice-dynamic-napt-44` supported for MS-MPC and MS-MIC
14.2R7	Dynamic Source NAT - NAPT44 Port Translation with Deterministic Port Block Allocation supported for MS-MPC and MS-MIC
14.2R7	IKE ALG
14.2R7	`deterministic-napt44` supported for MS-MPC and MS-MIC
14.2R7	Starting in Junos OS Release 14.2R7, 15.1R5, 16.1R2, and 17.1R1, the IKE ALG ALG is supported on MS-MPCs and MS-MICs.
14.2R2	Dynamic Source NAT - NAPT Port Translation with Secured Port Block Allocation supported for MS-MPC and MS-MIC
14.2	Starting In Junos OS 14.2, ICMP messages are handled by default and PING ALG support is provided.

NAT Configuration Overview

IN THIS SECTION

- Network Address Translation Configuration Overview | 92
- Configuring Source and Destination Addresses Network Address Translation Overview | 92
- Configuring Pools of Addresses and Ports for Network Address Translation Overview | 94
- Network Address Translation Rules Overview | 97
- Protecting CGN Devices Against Denial of Service (DOS) Attacks | 106
- Carrier-Grade NAT Implementation: Best Practices | 106

Network Address Translation Configuration Overview

To configure network address translation (NAT), complete the following high-level steps:

1. Configure the source and destination addresses. For more information, see "Configuring Source and Destination Addresses Network Address Translation Overview" on page 92 .
2. Define the addresses or prefixes, address ranges, and ports used for NAT. For more information, see "Configuring Pools of Addresses and Ports for Network Address Translation Overview" on page 94
3. If applicable, configure the address pools for network address port translation (NAPT). For more information, see "Configuring Address Pools for Network Address Port Translation (NAPT) Overview" on page 163 .
4. Configure the NAT rules. Within the rules, include match directions, match conditions, actions, and translation types. For more information, see "Network Address Translation Rules Overview" on page 97 .
5. Configure service sets for NAT processing. Within each service set, define the interfaces for handling inbound and outbound traffic and a NAT rule or ruleset. For more information, see Configuring Service Sets for Network Address Translation.

SEE ALSO

Carrier-Grade NAT Feature Comparison for Junos Address Aware by Type of Interface Card | 64

Configuring Source and Destination Addresses Network Address Translation Overview

You must configure a specific address, a prefix, or the address-range boundaries:

- The following addresses, while valid in `inet.0`, cannot be used for NAT translation:
  - 0.0.0.0/32
  - 127.0.0.0/8 (loopback)
  - 128.0.0.0/16 (martian)
  - 191.255.0.0/16 (martian)
  - 192.0.0.0/24 (martian)
  - 223.255.255.0/24 (martian)
  - 224.0.0.0/4 (multicast)
  - 240.0.0.0/4 (reserved)
  - 255.255.255.255 (broadcast)

  The addresses that are specified as valid in the `inet.0` routing table and not supported for NAT translation are `orlonger` match filter types. You cannot specify any regions within such address prefixes in a NAT pool.

- On MX Series routers with MS-MPCs and MS-MICs, if you configure a NAT address pool with a prefix length that is equal to or greater than /16, the PIC does not contain sufficient memory to provision the configured pool. Also, memory utilization problems might occur if you attempt to configure many pools whose combined total IP addresses exceed /16. In such circumstances, a system logging message is generated stating that the NAT pool name is failed to be created and that the service set is not activated. On MS-MPCs and MS-MICs, you must not configure NAT pools with prefix lengths greater than or equal to /16.

- You can specify one or more IPv4 address prefixes in the `pool` statement and in the `from` clause of the NAT rule term. This enables you to configure source translation from a private subnet to a public subnet without defining a rule term for each address in the subnet. Destination translation cannot be configured by this method. For more information, see *Examples: Configuring NAT Rules.*

- When you configure static source NAT, the `address` prefix size you configure at the `[edit services nat pool *pool-name*]` hierarchy level must be larger than the `source-address` prefix range configured at the `[edit services nat rule *rule-name* term *term-name* from]` hierarchy level. The `source-address` prefix range must also map to a single subnet or range of IPv4 or IPv6 addresses in the `pool` statement. Any pool addresses that are not used by the `source-address` prefix range are left unused. Pools cannot be shared.

- When you configure a NAT address pool prefix size with the `address` statement at the `[edit services nat pool *nat-pool-name*]` hierarchy level, the subnet and broadcast addresses are not included in the list of usable IP addresses. For example, if you use `address 10.11.12.0/28` in a NAT pool, the addresses 10.11.12.0 (subnet address) and 10.11.12.15 (broadcast address) are not available.

NOTE: When you include a NAT configuration that changes IP addresses, it might affect forwarding path features elsewhere in your router configuration, such as source class usage (SCU), destination class usage (DCU), filter-based forwarding, or other features that target specific IP addresses or prefixes.

NAT configuration might also affect routing protocol operation, because the protocol peering, neighbor, and interface addresses can be altered when routing protocols packets transit the Adaptive Services (AS) or Multiservices PIC.

SEE ALSO

Junos Address Aware Network Addressing Overview | 53

Configuring Pools of Addresses and Ports for Network Address Translation Overview

IN THIS SECTION

- Configuring NAT Pools | 94
- Preserve Range and Preserve Parity | 96
- Specifying Destination and Source Prefixes Without Configuring a Pool | 96

Configuring NAT Pools

You can use the `pool` statement to define the addresses (or prefixes), address ranges, and ports used for Network Address Translation (NAT). To configure the information, include the `pool` statement at the `[edit services nat]` hierarchy level.

Starting in Junos OS Release 14.2, configure the NAT pool as follows. Starting in Junos OS Release 16.1, the `limit-ports-per-address` statement is supported.

```
[edit services nat]
pool nat-pool-name {
    address ip-prefix</prefix-length>;
    address-range low minimum-value high maximum-value;
    limit-ports-per-address number;
    port {
        automatic (sequential | random-allocation);
        range low minimum-value high maximum-value random-allocation;
        preserve-parity;
        preserve-range {
    }
}
```

In Junos OS Release 14.1 and earlier, configure the NAT pool as follows:

```
[edit services nat]
pool nat-pool-name {
    address ip-prefix</prefix-length>;
    address-range low minimum-value high maximum-value;
    port (automatic | range low minimum-value high maximum-value);
        preserve-parity;
        preserve-range {
    }
}
```

To configure pools for traditional NAT, specify either a destination pool or a source pool.

With static source NAT and dynamic source NAT, you can specify multiple IPv4 addresses (or prefixes) and IPv4 address ranges. Up to 32 prefixes or address ranges (or a combination) can be supported within a single pool.

With static destination NAT, you can also specify multiple address prefixes and address ranges in a single term. Multiple destination NAT terms can share a destination NAT pool. However, the netmask or range for the `from` address must be smaller than or equal to the netmask or range for the destination pool address. If you define the pool to be larger than required, some addresses will not be used. For example, if you define the pool size as 100 addresses and the rule specifies only 80 addresses, the last 20 addresses in the pool are not used.

For constraints on specific translation types, see "Network Address Translation Rules Overview" on page 97 .

With source static NAT, the prefixes and address ranges cannot overlap between separate pools.

In an address range, the `low` value must be a lower number than the `high` value. When multiple address ranges and prefixes are configured, the prefixes are depleted first, followed by the address ranges.

When you specify a port for dynamic source NAT, address ranges are limited to a maximum of 65,000 addresses, for a total of (65,000 x 65,535) or 4,259,775,000 flows. A dynamic NAT pool with no address port translation supports up to 65,535 addresses. There is no limit on the pool size for static source NAT.

Preserve Range and Preserve Parity

You can configure your carrier-grade NAT (CGN) to preserve the range or parity of the packet source port when it allocates a source port for an outbound connection. You can configure the preserve parity and preserve range options under the NAT pool definition by including the `preserve-range` and `preserve-parity` configuration statements at the `[edit services nat pool` *`poolname`* `port]` hierarchy level.

Preserving range and preserving parity are supported on MX series routers with MS-DPCs and on M Series routers with MS-100, MS-400, and MS-500 MultiServices PICS. Preserving range and preserving parity are supported on MX series routers with MS-MPCs and MS-MICs starting in Junos OS release 15.1R1.

- Preserve range—RFC 4787, *Network Address Translation (NAT) Behavioral Requirements for Unicast UDP*, defines two ranges: 0 through 1023, and 1024 through 65,535. When the `preserve-range` knob is configured and the incoming port falls into one of these ranges, CGN allocates a port from that range only. However, if there is no available port in the range, the port allocation request fails and that session is not created. The failure is reflected on counters and system logging, but no Internet Control Message Protocol (ICMP) message is generated. If this knob is not configured, allocation is based on the configured port range without regard to the port range that contains the incoming port. The exception is some application-level gateways (ALGs), such as hello, that have special zones.

- Preserve parity—When the `preserve-parity` knob is configured, CGN allocates a port with the same even or odd parity as the incoming port. If the incoming port number is odd or even, the outgoing port number should correspondingly be odd or even. If a port number of the desired parity is not available, the port allocation request fails, the session is not created, and the packet is dropped.

Specifying Destination and Source Prefixes Without Configuring a Pool

You can directly specify the destination or source prefix used in NAT without configuring a pool.

To configure the information, include the `rule` statement at the `[edit services nat]` hierarchy level:

```
[edit services nat]
rule rule-name {
    term term-name {
        then {
            translated {
                destination-prefix prefix;
            }
        }
    }
}
```

Network Address Translation Rules Overview

IN THIS SECTION

- Configuring Match Direction for NAT Rules | 98
- Configuring Match Conditions in NAT Rules | 99
- Configuring Actions in NAT Rules | 100
- Configuring Translation Types | 102
- Configuring NAT Rules for IPsec Passthrough for Non-NAT-T Peers | 105

To configure a NAT rule, include the `rule` *`rule-name`* statement at the `[edit services nat]` hierarchy level:

```
[edit services nat]
allow-overlapping-nat-pools ;
    apply-groups;
    apply-groups-except;
    pool pool-name;
port-forwarding port-forwarding-name;
rule rule-name {
    match-direction (input | output);
    term term-name {
        from {
            application-sets set-name;
            applications [ application-names ];
            destination-address (address | any-unicast) <except>;
            destination-address-range low minimum-value high maximum-value <except>;
            destination-prefix-list list-name <except>;
            source-address (address | any-unicast) <except>;
            source-address-range low minimum-value high maximum-value <except>;
            source-prefix-list list-name <except>;
        }
        then {
            no-translation;
            translated {
                address-pooling paired;
                clat-prefix clat-prefix;
                destination-pool nat-pool-name;
                destination-prefix destination-prefix;
                dns-alg-pool dns-alg-pool;
                dns-alg-prefix dns-alg-prefix;
                filtering-type endpoint-independent;
                mapping-type endpoint-independent;
                overload-pool overload-pool-name;
                overload-prefix overload-prefix;
                source-pool nat-pool-name;
                source-prefix source-prefix;
                translation-type {
                    (basic-nat-pt | basic-nat44 | basic-nat66 | dnat-44 | dynamic-nat44 | napt-44 | napt-66 | napt-pt | stateful-nat464 | stateful-nat64 | twice-basic-nat-44 | twice-dynamic-nat-44 | twice-napt-44);
                }
            }
            syslog;
        }
    }
}
```

Each rule must include a `match-direction` statement that specifies the direction in which the match is applied.

NOTE: ACX Series routers support only *input* as the match direction.

In addition, each NAT rule consists of a set of terms, similar to a *firewall filter*. A term consists of the following:

- `from` statement—Specifies the match conditions and applications that are included and excluded.
- `then` statement—Specifies the actions and action modifiers to be performed by the router software.

The following sections explain how the components of NAT rules:

Configuring Match Direction for NAT Rules

Each rule must include a `match-direction` statement that specifies the direction in which the match is applied. To configure where the match is applied, include the `match-direction` statement at the `[edit services nat rule` *rule-name*`]` hierarchy level:

```
[edit services nat rule rule-name]
match-direction (input | output);
```

The match direction is used with respect to the traffic flow through the Multiservices DPC and Multiservices PICs. When a packet is sent to the PIC, direction information is carried along with it. The packet direction is determined based on the following criteria:

- With an interface service set, packet direction is determined by whether a packet is entering or leaving the interface on which the service set is applied.
- With a next-hop service set, packet direction is determined by the interface used to route the packet to the Multiservices DPC or Multiservices PIC. If the inside interface is used to route the packet, the packet direction is input. If the outside interface is used to direct the packet to the PIC or DPC, the packet direction is output. For more information about inside and outside interfaces, see "Configuring Service Sets to be Applied to Services Interfaces" on page 10 .
- On the Multiservices DPC and Multiservices PIC, a flow lookup is performed. If no flow is found, rule processing is performed. All rules in the service set are considered. During rule processing, the packet direction is compared against rule directions. Only rules with direction information that matches the packet direction are considered.

Configuring Match Conditions in NAT Rules

To configure NAT match conditions, include the `from` statement at the `[edit services nat rule *rule-name* term *term-name*]` hierarchy level:

```
[edit services nat rule rule-name term term-name]
from {
    application-sets set-name;
    applications [ application-names ];
    destination-address (address | any-unicast) <except>;
    destination-address-range low minimum-value high maximum-value <except>;
    destination-prefix-list list-name <except>;
    source-address (address | any-unicast) <except>;
    source-address-range low minimum-value high maximum-value <except>;
    source-prefix-list list-name <except>;
}
```

To configure traditional NAT, you can use the destination address, a range of destination addresses, the source address, or a range of source addresses as a match condition, in the same way that you would configure a firewall filter; for more information, see the Routing Policies, Firewall Filters, and Traffic Policers User Guide.

Alternatively, you can specify a list of source or destination prefixes by including the `prefix-list` statement at the `[edit policy-options]` hierarchy level and then including either the `destination-prefix-list` or source-prefix-list statement in the NAT rule. For an example, see "Examples: Configuring Stateful Firewall Rules" on page 557 .

If the translation-type statement in the then statement of the NAT rule is set to **stateful-nat-64**, the range specified by the destination-address-range or the destination-prefix-list in the from statement must be within the range specified by the destination-prefix statement in the then statement.

If at least one NAT term within a NAT rule has the address pooling paired (APP) functionality enabled (by including the address-pooling statement at the [edit services nat rule *rule-name* term *term-name* then translated] hierarchy level, all the other terms in the NAT rule that use the same NAT address pool as the address pool for the term with APP enabled must have APP enabled. Otherwise, if you add a NAT rule term without enabling APP to a rule that contains other terms with APP enabled, all the terms with APP enabled in a NAT rule drop traffic flows that match the specified criteria in the NAT rule.

For MX Series routers with MS-MICs and MS-MPCs, although the address pooling paired (APP) functionality is enabled within a NAT rule (by including the address-pooling statement at the [edit services nat rule *rule-name* term *term-name* then translated] hierarchy level), it is a characteristic of a NAT pool. Such a NAT pool for which APP is enabled cannot be shared with NAT rules that do not have APP configured.

When configuring NAT, if any traffic is destined for the following addresses and does not match a NAT flow or NAT rule, the traffic is dropped:

- Addresses specified in the from destination-address statement when you are using destination translation
- Addresses specified in the source NAT pool when you are using source translation

Configuring Actions in NAT Rules

To configure NAT actions, include the then statement at the [edit services nat rule *rule-name* term *term-name*] hierarchy level:

```
[edit services nat]
rule rule-name {
    term term-name {
        from {
            destination-address-range low minimum-value high maximum-value <except>;
            destination-prefix-list list-name <except>;
        }
        then {
            destination-prefix destination-prefix;
        }
```

```
[edit services nat rule rule-name term term-name]
then {
    no-translation;
    syslog;
    translated {
        clat-prefix clat-prefix;
        destination-pool nat-pool-name;
        destination-prefix destination-prefix;
        source-pool nat-pool-name;
        source-prefix source-prefix;
        translation-type (basic-nat-pt | basic-nat44 | basic-nat66 | dnat-44 | dynamic-nat44 |
napt-44 | napt-66 | napt-pt | stateful-nat464 | stateful-nat64 | twice-basic-nat-44 | twice-
dynamic-nat-44 | twice-napt-44);

        }
    }
}
```

- The `no-translation` statement allows you to specify addresses that you want excluded from NAT.

  The `no-translation` statement is supported on MX series routers with MS-DPCs and on M Series routers with MS-100, MS-400, and MS-500 MultiServices PICS. The `no-translation` statement is supported on MX series routers with MS-MPCs and MS-MICs starting in Junos OS release 15.1R1.

- The `system log` statement enables you to record an alert in the system logging facility.

- The `destination-pool`, `destination-prefix`, `source-pool`, and `source-prefix` statements specify addressing information that you define by including the `pool` statement at the `[edit services nat]` hierarchy level; for more information, see "Configuring Pools of Addresses and Ports for Network Address Translation Overview" on page 94 .

- The `translation-type` statement specifies the type of NAT used for source or destination traffic. The options are `basic-nat-pt`, `basic-nat44`, `basic-nat66`, `dnat-44`, `dynamic-nat44`, `napt-44`, `napt-66`, `napt-pt`, `stateful-nat464`, `stateful-nat64`, `twice-basic-nat-44`, `twice-dynamic-nat-44`, and `twice-napt-44`.

NOTE: In Junos OS Release 13.2 and earlier, the following restriction was not enforced by the CLI: if the `translation-type` statement in the `then` statement of a NAT rule was set to **stateful-nat-64**, the range specified by the `destination-address-range` or the `destination-prefix-list` in the `from` statement needed to be within the range specified by the destination-prefix statement in the then statement. Starting in Junos OS Release 13.3R1, this restriction is enforced.

Configuring Translation Types

The implementation details of the nine options of the translation-type statement are as follows:

- basic-nat44—This option implements the static translation of source IP addresses without port mapping. You must configure the from source-address statement in the match condition for the rule. The size of the address range specified in the statement must be the same as or smaller than the source pool. You must specify either a source pool or a destination prefix. The referenced pool can contain multiple addresses but you cannot specify ports for translation.

  NOTE: In an interface service set, all packets destined for the source address specified in the match condition are automatically routed to the services PIC, even if no service set is associated with the interface.

  NOTE: Prior to Junos OS Release 11.4R3, you could only use a source NAT pool in a single service set. As of Junos OS Release 11.4R3 and subsequent releases, you can reuse a source NAT pool in multiple service sets.

- basic-nat66—This option implements the static translation of source IP addresses without port mapping in IPv6 networks. The configuration is similar to the basic-nat44 implementation, but with IPv6 addresses.

  The basic-nat66 option is not available if you are using MS-MPCs or MS-MICs.

- basic-nat-pt—This option implements translation of addresses of IPv6 hosts, as they originate sessions to the IPv4 hosts in an external domain and vice versa. This option is always implemented with DNS ALG. You must define the source and destination pools of IPv4 addresses. You must configure one rule and define two terms. Configure the IPv6 addresses in the from statement in both term statements. In the then statement of the first term within the rule, reference both the source and destination pools and configure dns-alg-prefix. Configure the source prefix in the then statement of the second term within the same rule.

  The basic-nat-pt option is not available if you are using MS-MPCs or MS-MICs.

- deterministic-napt44—This option implements algorithm-based allocation of blocks of destination ports and IP address. This ensures that an incoming (source) IP address and port always map to the same destination IP address and port, thus eliminating the need for the address translation logging. When you use `deterministic-napt44`, you must also use `deterministic-port-block-allocation` at the `[edit services nat pool` *`poolname`* `port]` hierarchy level.

The `deterministic-napt44` option is supported on MX series routers with MS-DPCs and on M Series routers with MS-100, MS-400, and MS-500 MultiServices PICS. The `deterministic-napt44` option if you are using MX Series routers with MS-MPCs or MS-MICs is supported only in Junos OS release 14.2R7 and later 14.2 releases and in release 15.1R3 and later 15.1 releases.

- `dnat-44`—This option implements static translation of destination IP addresses without port mapping. The size of the pool address space must be greater than or equal to the destination address space. You must specify a name for the `destination pool` statement. The referenced pool can contain multiple addresses, ranges, or prefixes, as long as the number of NAT addresses in the pool is larger than the number of destination addresses in the `from` statement. You must include exactly one `destination-address` value at the `[edit services nat rule` *`rule-name`* `term` *`term-name`* `from]` hierarchy level; if it is a prefix, the size must be less than or equal to the pool prefix size. Any addresses in the pool that are not matched in the value remain unused, because a pool cannot be shared among multiple terms or rules.

- `dynamic-nat44`—This option implements dynamic translation of source IP addresses without port mapping. You must specify a `source-pool`. The referenced pool must include an `address` configuration (for address-only translation).

  The `dynamic-nat44` address-only option supports translating up to 16,777,216 addresses to a smaller size pool. The requests from the source address range are assigned to the addresses in the pool until the pool is used up, and any additional requests are rejected. A NAT address assigned to a host is used for all concurrent sessions from that host. The address is released to the pool only after all the sessions for that host expire. This feature enables the router to share a few public IP addresses between several private hosts. Because all the private hosts might not simultaneously create sessions, they can share a few public IP addresses.

- `napt-44`—This option implements dynamic translation of source IP addresses with port mapping. You must specify a name for the `source-pool` statement. The referenced pool must include a `port` configuration. If the port is configured as automatic or a port range is specified, then it implies that *Network Address Port Translation* (NAPT) is used.

- `napt-66`—This option implements dynamic address translation of source IP addresses with port mapping for IPv6 addresses. The configuration is similar to the `napt-44` implementation, but with IPv6 addresses.

  The `napt-66` option is not available if you are using MS-MPCs or MS-MICs.

- `napt-pt`—This option implements dynamic address and port translation for source and static translation of destination IP address. You must specify a name for the `source-pool` statement. The referenced pool must include a port configuration (for NAPT). Additionally, you must configure two rules, one for the DNS traffic and the other for the rest of the traffic. The rule meant for the DNS traffic should be DNS ALG enabled and the `dns-alg-prefix` statement should be configured. Moreover, the prefix configured in the `dns-alg-prefix` statement must be used in the second rule to translate the destination IPv6 addresses to IPv4 addresses.

The `napt-pt` option is not available if you are using MS-MPCs or MS-MICs.

- `stateful-nat464`—This option implements 464XLAT Provider-Side Translater (PLAT) address translation for source IP addresses and IPv6 prefix removal translation for destination IPv4 addresses. You must specify the IPv4 addresses used for translation at the [edit services nat pool] hierarchy level. This pool must be referenced in the rule that translates the IPv6 addresses to IPv4.

  The `stateful-nat464` option is available only if you are using MS-MPCs or MS-MICs, and is supported starting in Junos OS Release 17.1R1.

- `stateful-nat64`—This option implements dynamic address and port translation for source IP addresses and prefix removal translation for destination IP addresses. You must specify the IPv4 addresses used for translation at the `[edit services nat pool]` hierarchy level. This pool must be referenced in the rule that translates the IPv6 addresses to IPv4.

- `twice-basic-nat-44`—This option implements static source and static destination translation for IPv4 addresses, thus combining `basic-nat44` for source and `dnat-44` for destination addresses.

  The `twice-basic-nat-44` option is supported on MS-DPCs and MS-100, MS-400, and MS-500 MultiServices PICS. The `twice-basic-nat-44` option is supported on MS-MPCs and MS-MICs starting in Junos OS Release 15.1R1.

- `twice-dynamic-nat-44`—This option implements source dynamic and destination static translation for IPv4 addresses, combining `dynamic-nat44` for source and `dnat-44` for destination addresses.

  The `twice-dynamic-nat-44` option is supported on MS-DPCs and MS-100, MS-400, and MS-500 MultiServices PICS. The `twice-dynamic-nat-44` option is supported on MS-MPCs and MS-MICs starting in Junos OS Release 15.1R1.

- `twice-napt-44`—This option implements source NAPT and destination static translation for IPv4 addresses, combining `napt-44` for source and `dnat-44` for destination addresses.

  The `twice-napt-44` option is supported on MS-DPCs and MS-100, MS-400, and MS-500 MultiServices PICS. The `twice-napt-44` option is supported on MS-MPCs and MS-MICs starting in Junos OS Release 15.1R1.

For more information on NAT methods, see RFC 2663, *IP Network Address Translator (NAT) Terminology and Considerations*.

Configuring NAT Rules for IPsec Passthrough for Non-NAT-T Peers

Before Junos OS Release 17.4R1, Network Address Translation-Traversal (NAT-T) is not supported for the Junos VPN Site Secure suite of IPsec features on the MX Series routers. Starting in Junos OS Release 14.2R7, 15.1R5, 16.1R2, and 17.1R1, you can pass IKEv1 and IPsec packets through NAPT-44 and NAT64 rules between IPsec peers that are not NAT-T compliant. Only ESP tunnel mode is supported. This feature is supported only on MS-MPCs and MS-MICs.

To configure NAT rules for IPsec passthrough for NAPT-44 or NAT64:

1. Configure an IKE ALG application. See "Configuring Application Properties" on page 508 .
2. Add the application to an application set. See "Configuring Application Sets" on page 532 .
3. Configure a NAT pool. See "Configuring Pools of Addresses and Ports for Network Address Translation Overview" on page 94 .
4. Configure the NAT rule:
   a. Configure a match direction for the rule. See "Configuring Match Direction for NAT Rules" on page 98 .
   b. Configure one of the matching conditions to be the application set for IKE and IPsec passthrough that you configured in Step "2" on page 105 .

      ```
      [edit services nat rule rule-name term term-name from]
      user@host# set application-sets set-name
      ```

   c. Configure other match conditions. See "Configuring Match Conditions in NAT Rules" on page 99 .
   d. Configure the translation type as NAPT-44 or NAT64.

      ```
      [edit services nat rule rule-name term term-name then translated]
      user@host# set translation-type (napt-44 | stateful-nat64)
      ```

   e. Configure other NAT actions. See "Configuring Actions in NAT Rules" on page 100 .
5. Assign the NAT rule to a service set.

   ```
   [edit services]
   user@host# set service-set service-set-name nat-rules rule-name
   ```

SEE ALSO

Junos Address Aware Network Addressing Overview | 53

Protecting CGN Devices Against Denial of Service (DOS) Attacks

IN THIS SECTION

- Mapping Refresh Behavior | 106
- EIF Inbound Flow Limit | 106

You can now choose configuration options that help prevent or minimize the effect of attempted denial of service (DOS) attacks.

Mapping Refresh Behavior

Prior to the implementation of the new options for configuring NAT mapping refresh behavior, described in this topic, a conversation was kept alive when either inbound or outbound flows were active. This remains the default behavior. You can now also specify mapping refresh for only inbound flows or only outbound flows. To configure mapping refresh behavior, include the `mapping-refresh (inbound | outbound | inbound-outbound)` statement at the `[edit services nat rule` *`rule-name`* `term` *`term-name`* `then translated secure-nat-mapping]` hierarchy level.

EIF Inbound Flow Limit

Previously. the number of inbound connections on an EIF mapping was limited only by the maximum flows allowed on the system. You can now configure the number of inbound flows allowed for an EIF. To limit the number of inbound connections on an EIF mapping, include the `eif-flow-limit` *`number-of-flows`* statement at the `[edit services nat rule` *`rule-name`* `term` *`term-name`* `then translated secure-nat-mapping]` hierarchy level.

Carrier-Grade NAT Implementation: Best Practices

IN THIS SECTION

- Use Round-Robin Address-Allocation When Using APP with the MS-DPC | 107
- Use the EIM Feature Only When Needed | 108

- Define Port Block Allocation Block Sizes Based on Expected Number of User Sessions | **109**
- Considerations When Changing Port Block Allocation Configuration on Running Systems | **110**
- Do Not Allocate NAT Pools That Are Larger Than Needed | **111**
- Configure System Logging for NAT Only When Needed | **112**
- Limit the Impact of Missing IP Fragments | **114**
- Do Not Use Configurations Prone to Packet Routing Loops | **115**
- Inactivity Timeouts | **116**
- Enable Dump on Flow Control | **118**

The following topics present the best practices for carrier-grade NAT implementation:

Use Round-Robin Address-Allocation When Using APP with the MS-DPC

BEST PRACTICE: If you are using an MS-DPC and you configure address-pooling paired (APP) in a NAT rule, you should use round-robin address allocation for the NAT pool.

The APP feature maps a private IP address to the same public IP address in a NAT pool for all the NAT sessions for that private IP address.

Sequential address allocation for NAT pools is the default on the MS-DPC, and allocates all the ports for a public IP address before assigning the next IP address. Sequential allocation, together with APP, might result in mapping multiple private hosts to the same public IP address, resulting in fast port exhaustion for a public IP address while other ports are still available from the remaining IP addresses in the NAT pool.

Round-robin allocation, on the other hand, assigns the next IP address in the NAT pool to the next private IP address needing translation, reducing the chance that all the ports for one public IP address are depleted.

For more information about APP and round-robin address allocation, see "Configuring Address Pools for Network Address Port Translation (NAPT) Overview" on page 163 .

NOTE: The MS-MPC and MS-MIC only use round-robin allocation.

The following example shows round-robin address allocation.

```
[edit services]
nat pool natpool-1 {
    port {
        automatic;
    }
    address-allocation round-robin;
    mapping-timeout 120;
}
```

Use the EIM Feature Only When Needed

BEST PRACTICE: Do not use endpoint-independent mapping (EIM) in NAT rule terms that include Junos ALGs. EIM assigns the same external NAT address and port for a specific session from a private host, but adds processing overhead. EIM provides no benefit for any of the Junos ALGs, which already employ the functionality used by EIM.

BEST PRACTICE: Enable EIM for applications that do reuse the source ports and rely on a CGNAT device to maintain the same address and port mapping for all traffic sent to different destinations. For example, use EIM for console gaming applications such as Xbox and PS4 or applications that use unilateral self-address fixing methods (UNSAF). See (*IETF RFC 3424 IAB Considerations for Unilateral Self-Address Fixing (UNSAF) Across Network Address Translation*).

For more information about EIM, see "Configuring Address Pools for Network Address Port Translation (NAPT) Overview" on page 163 .

The following example uses the Junos SIP ALG in the NAT rule, so EIM is *not* used.

```
[edit services nat]
 rule natrule-1 {
    match-direction input;
    term1 {
        from {
            applications junos-sip;
        }
    }
```

```
        then {
            translated {
                source-pool natpool-3;
                translation-type {
                    napt-44;
                }
                address-pooling paired;
            }
        }
}
```

Define Port Block Allocation Block Sizes Based on Expected Number of User Sessions

BEST PRACTICE: For secure port block allocation and deterministic port block allocation, define a port block allocation block size that is 2 to 4 times larger than the expected average number of active sessions for a user. For example, if the user is expected to have an average of approximately 200 to 250 NAT sessions active, configuring the block size to 512 or 1024 provides a liberal allocation.

BEST PRACTICE: If you are rolling out secure port block allocation using the MX Series as a NAT device and are not sure of your subscriber user profile and traffic profile, set the port block size to 1024 if you have enough NAT IP addresses to handle the estimated peak number of private subscribers. The number of NAT IP addresses times 62 gives you the number of private subscribers that can be handled with a port block size of 1024 (there are 62 blocks per IP address). Then, closely monitor the MX Series router by using the `show services nat pool detail` command to determine whether the block size needs to be changed.

BEST PRACTICE: Be careful not to make the block size too large if the number of IP addresses you can allocate to the NAT pool is limited. Making a port block size that is large enough to efficiently assign the blocks to your subscribers might cause all the port blocks to be tied up.

Secure port block allocation allocates blocks of ports to a particular user for NAT44 or NAT64. Secure port block allocation limits the number of syslog messages by generating only one syslog per block of ports.

However, configuring the block size incorrectly can lead to inefficient use of NAT resources or to performance issues. For example, when a user connects to a website that requires the establishment of a significant number of sockets for a single HTML page, a corresponding number of new ports must be allocated. The port block size should be large enough to prevent continual allocation of new blocks. If the number of concurrent sessions for a private subscriber exceeds the number of ports available in the active port block, the other port blocks allocated to the subscriber are scanned for available ports to use or a new block is allocated from the free block pool for the subscriber. The scanning of allocated port blocks and allocation of additional blocks can result in delays in setting up new sessions and loading web pages.

For more information about port block allocation, see "Configuring Secured Port Block Allocation" on page 267 and "Configuring Deterministic NAPT" on page 197 .

The following example sets the port block size to 1024.

```
[edit services nat]
pool natpool-1 {
    address-range low 192.0.2.0 high 192.0.2.10;
    port {
        automatic;
        secure-port-block-allocation {
            block-size 1024;
             max-blocks-per-user 8;
            active-block-timeout 300;
        }
    }
    mapping-timeout 300;
}
```

Considerations When Changing Port Block Allocation Configuration on Running Systems

BEST PRACTICE: Before changing the secure port block allocation or deterministic port block configuration on a running system when using an MS-MPC or MS-MIC, plan for a quick disruption in the NAT sessions. The change in configuration results in the re-creation of all the current NAT sessions.

BEST PRACTICE: Before changing the port block allocation configuration on a running system when using an MS-DPC, plan for a disruption of services. After changing the configuration, you must reboot the MS-DPC, or if this is not possible, you must deactivate and reactivate the service set.

Changes to port block allocation configuration include:

- Changing any NAT pool PBA configuration.
- Changing a PBA NAT pool to a non-PBA NAT pool.
- Changing a non-PBA NAT pool to a PBA NAT pool.

For more information about configuring port block allocation, see "Configuring Secured Port Block Allocation" on page 267 and "Configuring Deterministic NAPT" on page 197 .

Do Not Allocate NAT Pools That Are Larger Than Needed

MS-MPC and MS-MIC

BEST PRACTICE: When using NAPT44 as your translation type with the MS-MIC or MS-MPC, do not configure NAT pools that are larger than needed for the peak session rate, which would tie up valuable IPv4 resources. Each conversation, also known as a session, includes two flows — an ingress and egress flow. Each conversation requires one port and each IP address in the pool has a 1024-65535 port range (64K), so the NAT pool size does not need to be larger than:

peak number of conversations /64K

BEST PRACTICE: When using NAPT44 as your translation type with the MS-MIC, we recommend a maximum NAT pool size of 128 addresses (a /25 network).

BEST PRACTICE: When using NAPT44 as your translation type with the MS-MPC, we recommend a maximum NAT pool size of 256 addresses (a /24 network).

The maximum recommended NAT pool size when using NAPT-44 for an MS-MIC is 128 IP addresses because the MS-MIC supports a maximum of 14 million flows, or 7 million conversations, which require 7 million ports. A total of 7 million ports are available with 128 IP addresses, with each IP address having a port range of 1024-65535.

The maximum recommended NAT pool size for each slot on an MS-MPC when using NAPT-44 is 256 IP addresses because each slot supports a maximum of 30 million flows, or 15 million conversations, which require 15 million ports. A total of 15 million ports are available with 256 IP addresses, with each IP address having a port range of 1024-65535.

You can use larger pools than the recommended values, and you can expect that configurations that use the port block allocation (PBA) feature require larger pools. This is because PBA assigns blocks of ports to private IP addresses, which changes the pool efficiency model.

For more information about configuring NAT pools, see "Configuring Pools of Addresses and Ports for Network Address Translation Overview" on page 94 .

MS-DPC

BEST PRACTICE: When using NAPT44 as your translation type with the MS-DPC, do not configure NAT pools that are larger than needed for the peak flow rate, which would tie up valuable IPv4 resources. Each conversation includes two flows (1 reverse flow for each forward flow). Each conversation requires one port and each IP address in the pool has a 1024-65535 port range (64K), so the NAT pool size does not need to be larger than:

peak number of conversations /64K

BEST PRACTICE: When using NAPT44 as your translation type with the MS-DPC, do not configure NAT pools with more than 64 addresses (a /26 network).

The maximum NAT pool size for an MS-DPC is 64 IP addresses because the MS-DPC supports a maximum of 8 million flows, or 4 million conversations, which requires a maximum of 4 million ports. A total of 4 million ports are available with 64 IP addresses, with each IP address having a port range of 1024-65535. If APP, EIM, and EIF are enabled, the MS-DPC supports a maximum of 5.8 million flows, or 2.9 million conversations, so the maximum NAT pool size would be less.

For more information about configuring NAT pools, see "Configuring Pools of Addresses and Ports for Network Address Translation Overview" on page 94 .

Configure System Logging for NAT Only When Needed

BEST PRACTICE: Do not enable system logging per session for secure port block allocation configurations.

BEST PRACTICE: Do not enable system logging for deterministic NAT configurations.

BEST PRACTICE: Enable system logging at the service-set level rather than at the services interface level when possible.

BEST PRACTICE: In production networks, always send the log messages to an external system log server. This avoids adding CPU load to the Routing Engine, which occurs when messages are logged locally.

BEST PRACTICE: Specify the system log class to restrict logging to the class of applications in which you are interested.

BEST PRACTICE: If you configure system logging within a NAT rule term, use a stateful firewall rule to restrict the traffic that reaches the NAT rule term.

System log messages can negatively affect the performance of the services card, depending on the frequency of creation and deletion of sessions. All system log messages created by the services card require CPU processing at the services card, and the system log messages themselves constitute traffic that is sent across the MX Series router and competes with user traffic to reach the external log server.

Secure port block allocation removes the need to configure logs per session, because you know the block and block size and can derive the ports allocated to each user.

Deterministic NAT removes the need to log at all, because all information on port allocation can be deduced mathematically.

The following example restricts logging to NAT events and sends log messages to the external log server 203.0.113.4

```
[edit services service-set S-SET-1]
class {
    nat-logs;
}
syslog {
    host 203.0.113.4;
}
```

When you configure system logging within a NAT rule term, all traffic that enters the NAT rule term generates a log, which can cause excessive logging. This might result in the logging rate limit being reached, and you would lose logs that you do need.

For more information about configuring system logging for NAT, see "Configuring NAT Session Logs" on page 347 .

Limit the Impact of Missing IP Fragments

BEST PRACTICE: For the services interface that is configured for NAT, limit the impact of missing or delayed fragments by configuring the following:

- Maximum number of fragments for a packet
- Maximum wait time for a missing fragment

IP fragments received by the services card configured for NAT are buffered as they arrive. This allows an integrity check of the completely reassembled packet before the packet is processed by NAT. Missing or delayed fragments can cause the already received fragments to be held until the internal buffer is full and they are flushed out, resulting in CPU usage overhead and reduced traffic forwarding.

Configuring the maximum number of fragments a packet can have and limiting the wait time for a missing fragment reduces the chance of the internal buffer becoming full.

The following example sets the maximum number of fragments to 10 and the maximum wait time to 3 seconds.

```
[edit interfaces ms-0/0/0]
services-options {
    fragment-limit 10;
    reassembly-timeout 3;
}
```

Do Not Use Configurations Prone to Packet Routing Loops

> **BEST PRACTICE**: Prevent packet routing loops by ensuring that only the intended traffic is allowed to reach the services card and be processed by the service set NAT rule. You can do this by:
>
> - Configuring a source-address range under the NAT rule when possible.
> - Configuring a firewall filter that accepts only the traffic meant to be serviced by the NAT rule in a next-hop style service set.

Packet looping between the Packet Forwarding Engine and the services card results in persistent high CPU usage on the services card. Packet looping can be caused by the services card receiving traffic from an unexpected private source network. When unexpected traffic is processed by NAT, a pinhole is created, and in the case of EIF many pinholes might be created. These pinholes cause routing loops if the return traffic routes back through the services card.

The following example shows a firewall filter that only allows traffic from 198.51.100.0/24 to reach services interface ms-1/0/0, which is the inside interface for a next-hop service set.

```
[edit firewall filter to_be_serviced]
term 1 {
    from }
        address {

        }
        198.51.100.0/24;
    }
    then accept;
}
term 2 {
    then disard;
    }
[edit interfaces ms-1/0/0]
    unit 1 {
        family intet {
            filter {
                output to_be_serviced;
            }
        }
        service-domain inside;
    }
```

For more information about configuring firewall filters, see the Routing Policies, Firewall Filters, and Traffic Policers User Guide.

The following example shows a NAT rule that only processes traffic from 198.51.100.0/24 (other traffic reaches the services interface, but is not processed).

```
[edit services nat]
rule rule_1 {
    match-direction input;
    term t1 {
        from {
            source-address {
                198.51.100.0/24;
            }

        }
        then {
            translated {
                source-pool pool1;
                translation-type {
                    napt-44;
                }
            }
        }

    }
}
```

For more information about configuring NAT rules, see "Network Address Translation Rules Overview" on page 97 .

Inactivity Timeouts

BEST PRACTICE: Set the inactivity timeout only for user-defined applications that could require the NAT session mapping to remain in memory for longer than the default NAT inactivity timeout of 30 seconds. For example, an HTTP or HTTPS banking application may require more than 30 seconds of inactivity because the user must enter data.

BEST PRACTICE: Before making changes to the existing inactivity timeouts, run the following commands several times during peak hours. Then run the commands after making the changes, and verify that the changes are not starving the MX Series router of NAT resources or the services card of memory.

- `show services sessions` count
- `show services nat pool` detail
- `show services service-sets summary`

The following example shows the inactivity timeout being set to 1800 seconds for HTTPS and HTTP applications.

```
[edit applications]
application https {
    inactivity-timeout 1800;
    destination-port 443;
    protocol tcp;
}
application http {
    inactivity-timeout 1800;
    destination-port 443;
    protocol tcp;
}
```

For more information about configuring user-defined applications, see "Configuring Application Properties" on page 508 .

You need to weigh the risks of setting high inactivity timeouts for all traffic. While the default NAT inactivity timeout of 30 seconds may be too low for some user-defined applications, setting a timeout value too high can tie up NAT resources. For example, setting high inactivity timeout values can tie up any TCP session that is inactive just minutes after it was created. If the TCP session is not cleanly closed by a FIN or RST by the client or server, the session will sit in memory and tie up the NAT resources assigned to it until the timeout value expires.

Setting higher inactivity timeouts that impact every UDP and TCP port can be dangerous, especially with UDP traffic like DNS. Unlike TCP, UDP has no way to end a session other than timing out, so all UDP sessions would stay active for the full inactivity timeout value.

The following example is *not* a recommended configuration because it sets high inactivity timeout values for all TCP and UDP traffic.

```
[edit applications]
application UDP-All {
     protocol UDP;
    source-port 1-65535;
     inactivity-timeout 3600;
}
application TCP-All {
     protocol TCP;
    source-port 1-65535;
     inactivity-timeout 3600;
}
```

We do not have specific recommended inactivity timeout values. The proper inactivity timeout values depend on several factors, including:

- What applications are used on an end user’s network

  For example, Apple has stated that an inactivity timeout of 60 minutes is required for the following Apple services, which require a long connection lifetime:

  - Apple Push Services: inbound TCP port 5223
  - Exchange Active Sync: inbound TCP port 443
  - MobileMe: inbound TCP ports 5222 and 5223
- How the NAT solution is being used, for example as a Gi NAT device or as an Enterprise edge router
- How large your NAT pools are
- How much traffic each services card receives during peak loads
- How much memory you have available

Enable Dump on Flow Control

> **BEST PRACTICE**: Enable the dump-on-flow-control option for any services card that is processing NAT traffic in a production network. This option detects when a services card is

locked up, writes a core dump that Juniper Networks can analyze to determine why the card locked up, and recovers the services card by restarting it.

For the MS-MIC and MS-MPC, set the dump-on-flow-control option under the pc- interface, which is used to send control traffic from the Routing Engine to the services card. The following example shows the configuration if the services interface is ms-2/1/0.

```
[edit interfaces pc-2/1/0]
 multiservice-options {
    flow-control-options {
         dump-on-flow-control;
    }
}
```

For the MS-DPC, set the dump-on-flow control option under the sp- interface. The following example shows the configuration if the services interface is sp-2/1/0.

```
[edit interfaces sp-2/1/0]
 multiservice-options {
    flow-control-options {
         dump-on-flow-control;
    }
}
```

SEE ALSO

Network Address Translation Configuration Overview | 92

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
17.1R1	Starting in Junos OS Release 14.2R7, 15.1R5, 16.1R2, and 17.1R1, you can pass IKEv1 and IPsec packets through NAPT-44 and NAT64 rules between IPsec peers that are not NAT-T compliant.
16.1	Starting in Junos OS Release 16.1, the limit-ports-per-address statement is supported.

14.2	Starting in Junos OS Release 14.2, configure the NAT pool as follows.

Network Address Translation Overview on ACX Series

IN THIS SECTION

- Network Address Translation Overview on ACX Series | 120
- Network Address Port Translation Overview | 121
- Network Address Translation Address Overload in ACX Series | 122
- Network Address Translation Constraints on ACX | 123
- Enabling Inline Services Interface on ACX Series | 124

Network Address Translation Overview on ACX Series

Network Address Translation (NAT) is a method for modifying or translating network address information in packet headers. Either or both source and destination addresses in a packet may be translated. NAT can include the translation of port numbers as well as IP addresses.

NAT is described in RFC 1631 to solve IP (version 4) address depletion problems. NAT has been found to be a useful tool for firewalls, traffic redirect, load sharing, network migrations, and so on.

NOTE: In ACX Series routers, NAT is supported only on the ACX1100 AC-powered router and ACX500 routers for inline NAT and inline IPsec services. ACX1100 AC-powered router supports only source NAT for IPv4 packets. Static and dynamic NAT types are currently not supported. Service chaining (GRE, NAT, and IPSec) on ACX1100-AC and ACX500 routers is not supported.

A license is required for enabling inline services on ACX500 routers.

NOTE: ACX5048 and ACX5096 routers do not support NAT configurations.

Source NAT is the translation of the source IP address of a packet leaving the router. Source NAT is used to allow hosts with private IP addresses to access a public network.

Source NAT allows connections to be initiated only for outgoing network connections—for example, from a private network to the Internet. Source NAT is commonly used to:

- Translate a single IP address to another address (for example, to provide a single device in a private network with access to the Internet).
- Translate a contiguous block of addresses to another block of addresses of the same size.
- Translate a contiguous block of addresses to another block of addresses of smaller size.
- Translate a contiguous block of addresses to a single IP address or a smaller block of addresses using port translation.
- Translate a contiguous block of addresses to the address of the egress interface.

SEE ALSO

Understanding Service Sets

Service Filters in ACX Series | **39**

Guidelines for Applying Service Filters | **40**

Service Filter Match Conditions for IPv4 Traffic | **42**

Service Filter Actions | **43**

CoS for NAT Services on ACX Series Routers

Configuring Address Pools for Network Address Port Translation (NAPT) Overview

Configuring Service Sets to Be Applied to Services Interfaces

Configuring Queuing and Scheduling on Inline Services Interface | **47**

Network Address Port Translation Overview

Network Address Port Translation (NAPT) is a method by which many network addresses and their TCP/UDP ports are translated into a single network address and its TCP/UDP ports. This translation can be configured in both IPv4 and IPv6 networks.

In ACX Series routers, you can have up to 4096 network address translations at a time.

SEE ALSO

Service Filters in ACX Series | **39**

Guidelines for Applying Service Filters | **40**

Service Filter Match Conditions for IPv4 Traffic | **42**

Service Filter Actions | 43

CoS for NAT Services on ACX Series Routers

Configuring Address Pools for Network Address Port Translation (NAPT) Overview

Configuring Service Sets to Be Applied to Services Interfaces

Configuring Queuing and Scheduling on Inline Services Interface | 47

Network Address Translation Address Overload in ACX Series

The NAT services on ACX Series routers allows Junos OS interface addresses to be shared with a NAPT pool. This feature of sharing the same address/port between the NAPT pool and Junos OS is termed as address overloading.

To achieve address overloading, the available IPv4 address or port range of 1 to 65,536 addresses is partitioned between Junos OS and NAT as shown below:

- Junos OS—1 to 49,159 addresses.
- NAPT pool—49,160 through 53,255 addresses.
- Junos OS—53,255 through 65,535 addresses.

The number of ports reserved for NAPT pool with address overload feature is 4096.

To enable address-overloading, include the `address-overload` statement and the `interface` statement at the `[edit services nat pool` *`nat-pool-name`*`]` hierarchy level.

The `address-overload` statement enables sharing of IPv4 address between Junos OS and the NAT pool. Along with the `address-overload` statement, you must also specify the `interface` statement so that the first available IPv4 address or port of the interface is picked up for the NAT pool.

You can configure the address overload feature the following ways:

- Configure an interface along with the `address-overload` statement as shown in the following example.

```
pool p3 {
    address-overload;
    interface ge-0/0/1.0;
    port {
        range low 49160 high 53255;
    }
}
```

  In this case, the primary address on the interface is picked for the NAT pool.

- Directly configure a /32 address as shown in the following example:

```
pool p4 {
    address-overload;
    address 45.0.0.1/32;
    port {
        range low 49160 high 53255;
    }
}
```

The `interface` statement enables sharing of IPv4 interface address with the NAT pool along with the port range specified in the pool.

SEE ALSO

- Understanding Service Sets
- Service Filters in ACX Series | 39
- Guidelines for Applying Service Filters | 40
- Service Filter Match Conditions for IPv4 Traffic | 42
- Service Filter Actions | 43
- *CoS for NAT Services on ACX Series Routers*
- Configuring Address Pools for Network Address Port Translation (NAPT) Overview
- Configuring Service Sets to Be Applied to Services Interfaces
- Configuring Queuing and Scheduling on Inline Services Interface | 47

Network Address Translation Constraints on ACX

You should consider the following constraints while configuring Network Address Translation (NAT) on ACX Series routers:

- When a port is defined in a NAT pool, you can configure only one address or one address range in the pool.

- ACX Series routers support `nat-rules` with `match-direction` as *input*. `match-direction` as *output* is not supported.

- When you specify an address range or an address prefix in a NAT pool, the maximum number of addresses supported is 65,535. ACX Series routers supports up to 4096 network address translations at a time.

- The maximum number of service sets that can be configured is 2.
- In a NAT rule term, the `from` clause can contain a maximum of 4 matching addresses.
- The maximum terms per NAT rule allowed is 4.
- The maximum NAT rules per service set allowed is 2.

SEE ALSO

Enabling Inline Services Interface on ACX Series | **124**

Understanding Service Sets

Guidelines for Applying Service Filters | **40**

CoS for NAT Services on ACX Series Routers

Configuring Address Pools for Network Address Port Translation (NAPT) Overview

Configuring Service Sets to Be Applied to Services Interfaces

Configuring Queuing and Scheduling on Inline Services Interface | **47**

Enabling Inline Services Interface on ACX Series

The inline services interface is a virtual interface that resides on the Packet Forwarding Engine. The `si-` interface makes it possible to provide NAT and IPsec services without using a special services PIC.

To configure inline services interface, you define the service interface as type `si-` (service-inline) interface. You must also reserve adequate bandwidth for the inline services interface. This enables you to configure both interface or next-hop service sets used for NAT and IPsec services.

NOTE: In ACX Series routers, you can configure only one inline services interface as an anchor interface for NAT and IPsec sessions: si-0/0/0.

NOTE: In ACX Series routers, only ACX1100-AC and ACX500 routers support IPsec services. ACX Series routers support only basic NAT.

To enable inline services interface:

1. Access an FPC-managed slot and the PIC where the interface is to be enabled.

```
[edit chassis]
user@host# edit fpc slot-number pic number
```

2. Enable the interface and specify the amount of bandwidth reserved on each Packet Forwarding Engine for tunnel traffic that uses inline services.

```
[edit chassis fpc slot-number pic number]
user@host# set inline-services bandwidth 1g
```

SEE ALSO

Network Address Translation Overview on ACX Series | **120**

Network Address Port Translation Overview | **121**

IPsec for ACX Series Overview | **640**

Understanding Service Sets

Service Filters in ACX Series | **39**

Guidelines for Applying Service Filters | **40**

Service Filter Match Conditions for IPv4 Traffic | **42**

Service Filter Actions | **43**

Network Address Translation Address Overload in ACX Series | **122**

CoS for NAT Services on ACX Series Routers

Network Address Translation Constraints on ACX | **123**

Configuring Address Pools for Network Address Port Translation (NAPT) Overview

Configuring Service Sets to Be Applied to Services Interfaces

Configuring Queuing and Scheduling on Inline Services Interface | **47**

CHAPTER 4

Stateful NAT64

IN THIS CHAPTER

- Stateful NAT64 | 126

Stateful NAT64

IN THIS SECTION

- Configuring Stateful NAT64 | 126

Configuring Stateful NAT64

To configure stateful NAT64, you must configure a rule at the `[edit services nat]` hierarchy level for translating the source address dynamically and the destination address statically.

> **BEST PRACTICE**: When you configure the service set that includes your NAT rule, include the `set stateful-nat64 clear-dont-fragment-bit` at the `[edit services service-set *service-set-name*]` hierarchy level.This clears the DF (don't fragment) bit in order to prevent unnecessary creation of an IPv6 fragmentation header when translating IPv4 packets that are less than 1280 bytes. RFC 6145, *IP/ICMP Translation Algorithm*, provides a full discussion of the use of the DF flag to control generation of fragmentation headers. For more information on service sets for NAT, see Configuring Service Sets for Network Address Translation.

To configure stateful NAT64:

1. In configuration mode, go to the `[edit services nat]` hierarchy level.

```
 [edit]
user@host# edit services nat
```

2. Define the pool of source addresses to be used for dynamic translation.

```
 [edit services nat]
user@host# set pool pool name address source addresses
user@host# set pool pool name port source ports
```

For example:

```
 [edit services nat]
user@host#  set pool src-pool-nat64 address 203.0.113.0/24
user@host#  set pool src-pool-nat64 port automatic
```

NOTE: Starting in Junos OS release 14.2, the `sequential` option is introduced to enable you to configure sequential allocation of ports. The `sequential` and `random-allocation` options available with the `port automatic` statement at the `[edit services nat pool` *`nat-pool-name`*`]` hierarchy level are mutually exclusive. You can include the `sequential` option for sequential allocation and the `random-allocation` option for random delegation of ports. By default, sequential allocation of ports takes place if you include only the `port automatic` statement at the `[edit services nat pool` *`nat-pool- name`*`]` hierarchy level. The `auto` option is hidden and is deprecated in Junos OS Release 14.2 and later, and is only maintained for backward compatibility. It might be removed completely in a future software release.

3. Define a NAT rule for translating the source addresses. Set the `match-direction` statement of the rule as **input**. Then define a term that uses **stateful-nat64** as the translation type for translating the addresses of the pool defined in the previous step.

```
 [edit services nat]
user@host# set rule rule name match-direction input
user@host# set rule rule name term term name from source-address source address
user@host# set rule rule name term term name from destination-address destination address
user@host# set rule rule name term term name then translated source-pool pool name
user@host# set rule rule name term term name then translated destination-prefix destination
```

```
prefix
user@host# set rule rule name term term name then translated translation-type stateful-nat64
```

For example:

```
 [edit services nat]
user@host# set rule stateful-nat64 match-direction input
user@host# set rule stateful-nat64 term t1 from source-address 2001:DB8::0/96
user@host# set rule stateful-nat64 term t1 from destination-address 64:FF9B::/96
user@host# set rule stateful-nat64 term t1 then translated source-pool src-pool-nat64
user@host# set rule stateful-nat64 term t1 then translated destination-prefix 64:FF9B::/96
user@host# set rule stateful-nat64 term t1 then translated translation-type stateful-nat64
```

The following example configures dynamic source address (IPv6-to-IPv4) and static destination address (IPv6-to-IPv4) translation.

```
[edit services]
user@host# show
nat {
    pool src-pool-nat64 {
        address 203.0.113.0/24;
        port {
            automatic;
        }
    }
    rule stateful-nat64 {
        match-direction input;
        term t1 {
            from {
                source-address {
                    2001:db8::0/96;
                }
                destination-address {
                    64:ff9b::/96;
                }
            }
            then {
                translated {
                    source-pool src-pool-nat64;
                    destination-prefix 64:ff9b::/96;
                    translation-type {
                        stateful-nat64;
                    }
                }
            }
        }
    }
}
service-set sset-nat64 {
    nat-options {
        stateful-nat64 {
            clear-dont-fragment-bit;
        }
    }
    service-set-options;
    nat-rules stateful-nat64;
    interface-service {
        service-interface ms-0/1/0;
    }
}
```

NOTE: If you configure two NAT64 rules and associate them with the same service set, along with stateful firewall rules, and apply the service set on two VLAN-tagged interfaces, for traffic that is transmitted matching both the NAT rules, the traffic that is destined to the second NAT rule is dropped. In such a scenario, traffic flows are not dropped on the Routing Engine. This behavior of traffic drop by the second NAT rule is expected. With Junos OS Extension-Provider packages installed on a device, because endpoint-independent mapping (EIM) is not supported, EIM per VLAN or per NAT rule term. The second session, which is dropped by the second NAT rule in the configuration scenario described here, is not created owing to the following sequence of events:

1. The first packet matching either rule creates an EIM and a session.

2. The second packet matches the EIM entry because the second packet is sent with the same source IP address and port as the first packet (but with a different destination address).

This condition causes allocation (reuse) of the same public IP address and port to the second packet as the first packet. The reverse flow for this session has the same 5-tuple data as the reverse flow of the first session. This behavior causes flow addition failure because a duplicate flow in the same service set is not permitted.

To work around this problem, disable EIM in both the NAT rules, which causes both the sessions to be established and processed correctly. Alternatively, to avoid this problem, specify the NAT rules on different service-sets configured on different units of the media interface with EIM enabled to successfully establish both the sessions.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
14.2	Starting in Junos OS release 14.2, the `sequential` option is introduced to enable you to configure sequential allocation of ports.

CHAPTER 5

Static Source NAT

IN THIS CHAPTER

- Static Source NAT | 131

Static Source NAT

IN THIS SECTION

- Configuring Static Source Translation in IPv4 Networks | 131
- Configuring Static Source Translation in IPv6 Networks | 141
- Example: Configuring Basic NAT44 | 147
- Example: Configuring NAT for Multicast Traffic | 150

Configuring Static Source Translation in IPv4 Networks

IN THIS SECTION

- Configuring the NAT Pool and Rule | 132
- Configuring the Service Set for NAT | 135
- Configuring Trace Options | 137
- Sample Configuration - Static Source NAT Using a Static Pool With An Address Prefix And An Address Range | 139
- Sample Configuration - Static Source Nat for One-To-One Mapping Between a Private Subnet and a Public Subnet | 139

To configure the translation type as **basic-nat44**, you must configure the NAT pool and rule, service set with service interface, and trace options. This topic includes the following tasks:

Configuring the NAT Pool and Rule

To configure the NAT pool, rule, and term:

1. In configuration mode, go to the `[edit services nat]` hierarchy level.

```
 [edit]
user@host# edit services nat
```

2. Configure the NAT pool with an address.

```
 [edit services nat]
user@host# set pool pool name address address
```

   In the following example, the pool name is **src_pool** and the address is **10.10.10.2/32**.

```
 [edit services nat]
user@host# set pool src_pool address 10.10.10.2/32
```

3. Configure the NAT rule and the match direction.

```
 [edit services nat]
user@host# set rule rule-name match-direction match-direction
```

   In the following example, the NAT rule name is **rule-basic-nat44** and the match direction is **input**.

```
 [edit services nat]
user@host# set rule rule-basic-nat44 match-direction input
```

4. Configure the source address in the `from` statement.

```
 [edit services nat]
user@host# set rule rule-basic-nat44 term term-name from from source-address address
```

In the following example, the term name is **t1** and the input condition is **source-address 3.1.1.2/32**.

```
 [edit services nat]
user@host# set rule rule-basic-nat44 term t1 from source-address 3.1.1.2/32
```

5. Configure the NAT term action and properties of the translated traffic.

```
 [edit services nat]
user@host# set rule rule-basic-nat44 term t1 then term-action translated-property
```

In the following example, the term action is **translated** and the property of the translated traffic is **source-pool src_pool**.

```
 [edit services nat]
user@host# set rule rule-basic-nat44 term t1 then translated source-pool src_pool
```

6. Configure the translation type.

```
 [edit services nat]
user@host# set rule rule-basic-nat44 term t1 then translated translation-type translation-type
```

In the following example, the translation type is **basic-nat44**.

```
 [edit services nat]
user@host# set rule rule-basic-nat44 term t1 then translated translation-type basic-nat44
```

7. Verify the configuration by using the `show` command at the `[edit services nat]` hierarchy level.

```
[edit services]
user@host# show
nat {
    pool src_pool {
        address 10.10.10.2/32;
    }
    rule rule-basic-nat44 {
        match-direction input;
        term t1 {
            from {
                source-address {
                    3.1.1.2/32;
                }
            }
            then {
                translated {
                    source-pool src_pool;
                    translation-type {
                        basic-nat44;
                    }
                }
            }
        }
    }
}
```

NOTE: If you don’t configure a stateful firewall (SFW) rule for your traffic, then each packet is subjected to the following default stateful firewall rule:

- Allow any valid packets from inside to outside.
- Create forward and return flow based on packets 5-tuple.
- Allow only valid packets matching return flows from outside to inside.

The stateful firewall’s packet validity checks are described in the *Stateful Firewall Anomaly Checking* in "Junos Network Secure Overview" on page 546 . When a packets pass stateful firewall validity checking but are not matched by a NAT rule, they are not translated and may be forwarded if the NAT node has a valid route to the packets’ destination IP addresses.

NOTE: When you add or delete a parameter in the `from` statement (NAT rule term match condition) at the `[edit services service-set service-set-name nat-rules` *`rule-name`* `term` *`term- name`*`]` hierarchy level, this configuration change triggers a deletion and addition of the NAT policy (which is equivalent to the deactivation and activation of a service set) that causes all existing NAT mappings to be deleted. Because the sessions are not closed owing to the change in the NAT policy, this behavior causes the mappings to timeout immediately after the sessions are closed. This behavior is expected and is applicable only with Junos OS Extension-Provider packages installed on a device. When a NAT policy is deleted and readded, only EIM mappings are deleted. This NAT policy change does not deactivate and activate the service set. We recommend that you deactivate and reactivate the service set in such scenarios in Junos OS Release 14.2 and earlier.

Configuring the Service Set for NAT

To configure the service set for NAT:

1. In configuration mode, go to the `[edit services]` hierarchy level.

   ```
    [edit]
   user@host# edit services
   ```

2. Configure the service set.

   ```
    [edit services]
   user@host# edit service-set service-set-name
   ```

   In the following example, the service set name is **s1**.

   ```
    [edit services]
   user@host# edit service-set s1
   ```

3. For the **s1** service set, set the reference to the NAT rules configured at the `[edit services nat]` hierarchy level.

   ```
    [edit services service-set s1]
   user@host# set nat-rules rule-name
   ```

   In the following example, the rule name is **rule-basic-nat44**.

   ```
    [edit services service-set s1]
   user@host# set nat-rules rule-basic-nat44
   ```

4. Configure the service interface.

   ```
    [edit services service-set s1]
   user@host# set interface-service service-interface service-interface-name
   ```

In the following example, the service interface name is **ms-1/2/0**.

```
 [edit services service-set s1]
user@host# set interface-service service-interface ms-1/2/0
```

NOTE: If you have a Trio-based line card, you can configure an inline-services interface on that card:

```
[edit]
user@host# set interfaces si-0/0/0
 [edit services service-set s1]
user@host# set interface-service service-interface si-0/0/0
```

5. Verify the configuration by using the `show` command at the `[edit services]` hierarchy level.

```
[edit services]
user@host# show
service-set s1 {
    nat-rules rule-basic-nat44;
    interface-service {
        service-interface ms-1/2/0;
    }
}
```

6. Associate the NAT service set with an `xe-` interface:

```
user@host# set interfaces xe-1/1/0 unit 0 family inet address 10.255.247.2/24
user@host# set interfaces xe-1/1/0 unit 0 family inet service input service-set s1
user@host# set interfaces xe-1/1/0 unit 0 family inet service output service-set s1
```

7. Verify the configuration by using the `show` command at the `[edit interfaces]` hierarchy level.

```
[edit interfaces]
user@host# show
xe-1/1/0 {
    unit 0 {
        family inet {
            service {
                input {
                    service-set s1;
                }
                output {
                    service-set s1;
                }
             }
            address 10.255.247.2/24;
        }
    }
}
```

Configuring Trace Options

To configure the trace options:

1. In configuration mode, go to the `[edit services adaptive-services-pics]` hierarchy level.

```
[edit]
user@host# edit services adaptive-services-pics
```

2. Configure the trace options.

```
[edit services adaptive-services-pics]
user@host# set traceoptions flag tracing parameter
```

   In the following example, the tracing parameter is **all**.

```
[edit services adaptive-services-pics]
user@host# set traceoptions flag all
```

3. Verify the configuration by using the `show` command at the `[edit services]` hierarchy level.

```
[edit services]
user@host# show
adaptive-services-pics {
    traceoptions {
        flag all;
    }                                    }
```

```
[edit]
user@host# show services
service-set s1 {
    nat-rules rule-basic-nat44;
    interface-service {
        service-interface ms-1/2/0;
    }
}
nat {
    pool src_pool {
        address 10.10.10.2/32;
    }
    rule rule-basic-nat44 {
        match-direction input;
        term t1 {
            from {
                source-address {
                    3.1.1.2/32;
                }
            }
            then {
                translated {
                    source-pool src_pool;
                    translation-type {
                        basic-nat44;
                    }
                }
            }
        }
    }
}
adaptive-services-pics {
    traceoptions {
        flag all;
    }
}
```

Sample Configuration - Static Source NAT Using a Static Pool With An Address Prefix And An Address Range

```
[edit services nat]
pool p1 {
    address 30.30.30.252/30;
    address-range low 20.20.20.1 high 20.20.20.2;
}
rule r1 {
    match-direction input;
    term t1 {
        from {
            source-address {
                10.10.10.252/30;
            }
        }
        then {
            translated {
                source-pool p1;
                translation-type basic-nat44;
            }
        }
    }
}
```

Sample Configuration - Static Source Nat for One-To-One Mapping Between a Private Subnet and a Public Subnet

```

[edit]
user@host# show services
service-set s1 {
    nat-rules rule-basic-nat44;
    interface-service {
        service-interface ms-1/2/0;
    }
}
nat {
    pool src_pool {
        address 10.10.10.2/32;
    }
    rule rule-basic-nat44 {
        match-direction input;
        term t1 {
            from {
                source-address {
                    3.1.1.2/32;
                }
            }
            then {
                translated {
                    source-pool src_pool;
                    translation-type {
                        basic-nat44;
                    }
                }
            }
        }
    }
}
adaptive-services-pics {
    traceoptions {
        flag all;
    }
}
```

```
[edit interfaces]
user@host# show
xe-1/1/0 {
    unit 0 {
        family inet {
            service {
                input {
                    service-set s1;
                }
                output {
                    service-set s1;
                }
             }
            address 10.255.247.2/24;
        }
    }
}
```

Configuring Static Source Translation in IPv6 Networks

IN THIS SECTION

- Configuring the NAT Pool and Rule | 141
- Configuring the Service Set for NAT | 143
- Configuring Trace Options | 145

To configure the translation type as basic-nat66, you must configure the NAT pool and rule, service set with service interface, and trace options. The basic-nat66 translation type is not available if you are using MS-MPCs or MS-MICs.

This topic includes the following tasks:

Configuring the NAT Pool and Rule

To configure the NAT pool, rule, and term:

1. In configuration mode, go to the [edit services nat] hierarchy level.

```
[edit]
user@host# edit services nat
```

2. Configure the NAT pool with an address.

```
 [edit services nat]
user@host# set pool pool name address address
```

In the following example, the pool name is **src_pool** and the address is **10.10.10.2/32**.

```
 [edit services nat]
user@host# set pool src_pool address 10.10.10.2/32
```

3. Configure the NAT rule and the match direction.

   ```
    [edit services nat]
   user@host# set rule rule-name match-direction match-direction
   ```

   In the following example, the rule name is **rule-basic-nat66** and the match direction is **input**.

   ```
    [edit services nat]
   user@host# set rule rule-basic-nat66 match-direction input
   ```

4. Configure the source address in the `from` statement.

   ```
    [edit services nat]
   user@host# set rule rule-basic-nat66 term term-name from from source-address address
   ```

   In the following, the term name is **t1** and the input condition is **source-address 2001:db8:10::0/96**.

   ```
    [edit services nat]
   user@host# set rule rule-basic-nat66 term t1 from source-address 2001:db8:10::0/96
   ```

5. Configure the NAT term action and properties of the translated traffic.

   ```
    [edit services nat]
   user@host# set rule rule-basic-nat66 term t1 then term-action translated-property
   ```

   In the following example, the term action is **translated** and the property of the translated traffic is **source-pool src_pool**.

   ```
    [edit services nat]
   user@host# set rule rule-basic-nat66 term t1 then translated source-pool src_pool
   ```

6. Configure the translation type.

   ```
    [edit services nat]
   user@host# set rule rule-basic-nat66 term t1 then translated translation-type translation-type
   ```

In the following example, the translation type is **basic-nat66**.

```
 [edit services nat]
user@host# set rule rule-basic-nat66 term t1 then translated translation-type basic-nat66
```

7. Verify the configuration by using the `show` command at the `[edit services]` hierarchy level.

```
[edit services]
user@host# show
nat {
    pool src_pool {
        address 10.10.10.2/32;
    }
    rule rule-basic-nat66 {
        match-direction input;
        term t1 {
            from {
                source-address {
                    2001:db8:10::0/96;
                }
            }
            then {
                translated {
                    source-pool src_pool;
                    translation-type {
                        basic-nat66;
                    }
                }
            }
        }
    }
}
```

Configuring the Service Set for NAT

To configure the service set for NAT:

1. In configuration mode, go to the `[edit services]` hierarchy level.

   ```
   [edit]
   user@host# edit services
   ```

2. Configure the service set.

   ```
    [edit services]
   user@host# edit service-set service-set-name
   ```

   In the following example, the service set name is **s1**.

   ```
    [edit services]
   user@host# edit service-set s1
   ```

3. For the **s1** service set, set the reference to the NAT rules configured at the `[edit services nat]` hierarchy level.

   ```
    [edit services service-set s1]
   user@host# set nat-rules rule-name
   ```

   In the following example, the rule name is **rule-basic-nat66**.

   ```
    [edit services service-set s1]
   user@host# set nat-rules rule-basic-nat66
   ```

4. Configure the service interface.

   ```
    [edit services service-set s1]
   user@host# set interface-service service-interface service-interface-name
   ```

   In the following example, the service interface name is **sp-1/2/0**.

   ```
    [edit services service-set s1]
   user@host# set interface-service service-interface sp-1/2/0
   ```

5. Verify the configuration by using the `show` command at the `[edit services]` hierarchy level.

```
[edit services]
user@host# show
service-set s1 {
    nat-rules rule-basic-nat66;
    interface-service {
        service-interface sp-1/2/0;
    }
}
```

Configuring Trace Options

To configure the trace options at the `[edit services adaptive-services-pics]` hierarchy level:

1. In configuration mode, go to the `[edit services adaptive-services-pics]` hierarchy level.

```
[edit]
user@host# edit services adaptive-services-pics
```

2. Configure the trace options.

```
[edit services adaptive-services-pics]
user@host# set traceoptions flag tracing parameter
```

   In the following example, the tracing parameter is **all**.

```
[edit services adaptive-services-pics]
user@host# set traceoptions flag all
```

3. Verify the configuration by using the `show` command at the `[edit services]` hierarchy level.

```
[edit services]
user@host# show
adaptive-services-pics {
    traceoptions {
        flag all;
    }
}
```

The following example configures the translation type as **basic-nat66**.

```
[edit]
user@host# show services
service-set s1 {
    nat-rules rule-basic-nat66;
    interface-service {
        service-interface sp-1/2/0;
    }
}
nat {
    pool src_pool {
        address 10.10.10.2/32;
    }
    rule rule-basic-nat66 {
        match-direction input;
        term t1 {
            from {
                source-address {
                    2001:db8:10::0/96/96;
                }
            }
            then {
                translated {
                    source-pool src_pool;
                    translation-type {
                        basic-nat66;
                    }
                }
            }
        }
    }
}
adaptive-services-pics {
    traceoptions {
        flag all;
    }
}
```

Example: Configuring Basic NAT44

IN THIS SECTION

- Requirements | 147
- Overview | 147
- Configuring Basic NAT44 | 147

This example describes how to implement a basic NAT44 configuration.

Requirements

This example uses the following hardware and software components:

- An MX Series 5G Universal Routing Platform with a Services DPC or an M Series Multiservice Edge router with a services PIC
- A domain name server (DNS)
- Junos OS Release 11.4 or higher

Overview

This example shows a complete CGN NAT44 configuration and advanced options.

Configuring Basic NAT44

IN THIS SECTION

- Chassis Configuration | 148
- Interfaces Configuration | 148

Chassis Configuration

Step-by-Step Procedure

To configure the service PIC (FPC 5 Slot 0) with the Layer 3 service package:

1. Go to the **[edit chassis]** hierarchy level.

```
user@host# edit chassis
```

2. Configure the Layer 3 service package.

```
[edit chassis]
user@host# set fpc 5 pic 0 adaptive-services service-package layer-3
```

Interfaces Configuration

Step-by-Step Procedure

To configure interfaces to the private network and the public Internet:

1. Define the interface to the private network.

```
user@host# edit interfaces ge-1/3/5
[edit interfaces ge-1/3/5]
user@host# set description “Private”
user@host# edit unit 0 family inet
[edit interfaces ge-1/3/5 unit 0 family inet]
user@host# set service input service-set ss2
user@host# set service output service-set ss2
user@host# set address 9.0.0.1/24
```

2. Define the interface to the public Internet.

```
user@host# edit interfaces ge-1/3/6
[edit interfaces ge-1/3/6]
user@host# set description “Public”
user@host# set unit 0 family inet address 128.0.0.1/24
```

3. Define the service interface for NAT processing.

   ```
   user@host# edit interfaces sp-5/0/0
   [edit interfaces sp-5/0/0]
   user@host# set unit 0 family inet
   ```

Results

```
user@host# show interfaces ge-1/3/5
description Private;
unit 0 {
    family inet {
        service {
            input {
                 service-set sset2;
            }
            output {
                service-set sset2;
            }
        }
        address 9.0.0.1/24;
    }
}
}
```

```
user@host# show interfaces ge-1/3/6
description Public:;
unit 0 {
    family inet {
        address 128.0.0.1/24;
    }
}
```

```
user@host# show interfaces sp-5/0/0
unit 0 {
family inet;
}
```

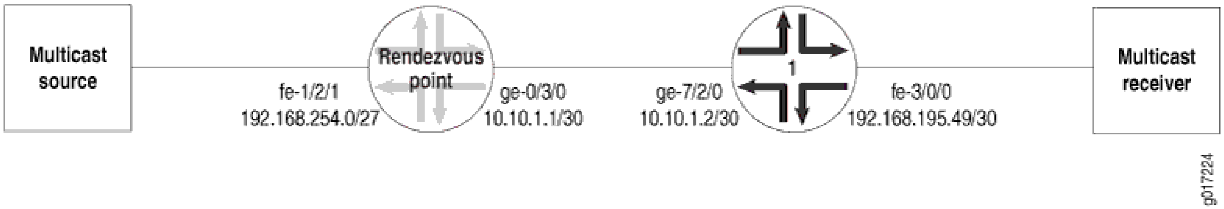
Example: Configuring NAT for Multicast Traffic

IN THIS SECTION

- Rendezvous Point Configuration | 150
- Router 1 Configuration | 154

Figure 9 on page 150 illustrates the network setup for the following configuration, which allows IP multicast traffic to be sent to the Multiservices PIC.

Figure 9: Configuring NAT for Multicast Traffic



Rendezvous Point Configuration

On the rendezvous point (RP), all incoming traffic from the multicast source at **192.168.254.0/27** is sent to the static NAT pool **mcast_pool**, where its source is translated to **20.20.20.0/27**. The service set **nat_ss** is a next-hop service set that allows IP multicast traffic to be sent to the Multiservices DPC or Multiservices PIC. The inside interface on the PIC is **ms-1/1/0.1** and the outside interface is **ms-1/1/0.2**.

```
[edit services]
nat {
    pool mcast_pool {
        address 20.20.20.0/27;
    }
    rule nat_rule_1 {
        match-direction input;
        term 1 {
            from {
                source-address 192.168.254.0/27;
            }
        }
        then {
            translated {
                source-pool mcast_pool;
                translation-type basic-nat44;
            }
            syslog;
        }
    }
}
service-set nat_ss {
    allow-multicast;
    nat-rules nat_rule_1;
    next-hop-service {
        inside-service-interface ms-1/1/0.1;
        outside-service-interface ms-1/1/0.2;
    }
}
```

The Gigabit Ethernet interface **ge-0/3/0** carries traffic out of the RP to Router 1. The multiservices interface **ms-1/1/0** has two logical interfaces: **unit 1** is the inside interface for next-hop services and **unit 2** is the outside interface for next-hop services. Multicast source traffic comes in on the Fast Ethernet interface **fe-1/2/1**, which has the firewall filter **fbf** applied to incoming traffic.

```
[edit interfaces]
ge-0/3/0 {
    unit 0 {
        family inet {
            address 10.10.1.1/30;
        }
    }
}
ms-1/1/0 {
    unit 0 {
        family inet;
    }
    unit 1 {
        family inet;
        service-domain inside;
    }
    unit 2 {
        family inet;
        service-domain outside;
    }
}
fe-1/2/1 {
    unit 0 {
        family inet {
            filter {
                input fbf;
            }
            address 192.168.254.27/27;
        }
    }
}
```

Multicast packets can only be directed to the Multiservices DPC or the Multiservices PIC using a next-hop service set. In the case of NAT, you must also configure a VPN routing and forwarding instance (VRF). Therefore, the routing instance **stage** is created as a “dummy” forwarding instance. To direct incoming packets to **stage**, you configure filter-based forwarding through a firewall filter called **fbf**, which is applied to the incoming interface **fe-1/2/1**. A lookup is performed in **stage.inet.0**, which has a multicast static route that is installed with the next hop pointing to the PIC’s inside interface. All multicast traffic matching this route is sent to the PIC.

```
[edit firewall]
filter fbf {
    term 1 {
        then {
            routing-instance stage;
        }
    }
}
```

The routing instance **stage** forwards IP multicast traffic to the inside interface **ms-1/1/0.1** on the Multiservices DPC or Multiservices PIC:

```
[edit]
routing-instances stage {
    instance-type forwarding;
    routing-options {
        static {
            route 224.0.0.0/4 next-hop ms-1/1/0.1;
        }
    }
}
```

You enable OSPF and Protocol Independent Multicast (PIM) on the Fast Ethernet and Gigabit Ethernet logical interfaces over which IP multicast traffic enters and leaves the RP. You also enable PIM on the outside interface (**ms-1/1/0.2**) of the next-hop service set.

```
[edit protocols]
ospf {
    area 0.0.0.0 {
        interface fe-1/2/1.0 {
            passive;
        }
        interface lo0.0;
        interface ge-0/3/0.0;
    }
}
pim {
    rp {
        local {
            address 10.255.14.160;
        }
    }
    interface fe-1/2/1.0;
    interface lo0.0;
    interface ge-0/3/0.0;
    interface ms-1/1/0.2;
}
```

As with any filter-based forwarding configuration, in order for the static route in the forwarding instance **stage** to have a reachable next hop, you must configure routing table groups so that all interface routes are copied from **inet.0** to the routing table in the forwarding instance. You configure routing tables **inet.0** and **stage.inet.0** as members of **fbf_rib_group**, so that all interface routes are imported into both tables.

```
[edit routing-options]
interface-routes {
    rib-group inet fbf_rib_group;
}
rib-groups fbf_rib_group {
    import-rib [ inet.0 stage.inet.0 ];
}
```

```
multicast {
    rpf-check-policy no_rpf;
}
```

Reverse path forwarding (RPF) checking must be disabled for the multicast group on which source NAT is applied. You can disable RPF checking for specific multicast groups by configuring a policy similar to the one in the example that follows. In this case, the **no_rpf** policy disables RPF check for multicast groups belonging to **224.0.0.0/4**.

```
[edit policy-options]
policy-statement no_rpf {
    term 1 {
        from {
            route-filter 224.0.0.0/4 orlonger;
        }
        then reject;
    }
}
```

Router 1 Configuration

The Internet Group Management Protocol (IGMP), OSPF, and PIM configuration on Router 1 is as follows. Because of IGMP static group configuration, traffic is forwarded out **fe-3/0/0.0** to the multicast receiver without receiving membership reports from host members.

```
[edit protocols]
igmp {
    interface fe-3/0/0.0 {
    }
}
ospf {
    area 0.0.0.0 {
        interface fe-3/0/0.0 {
            passive;
        }
        interface lo0.0;
        interface ge-7/2/0.0;
    }
    pim {
        rp {
            static {
                address 10.255.14.160;
            }
        }
        interface fe-3/0/0.0;
        interface lo0.0;
        interface ge-7/2/0.0;
    }
}
```

The routing option creates a static route to the NAT pool, **mcast_pool**, on the RP.

```
[edit routing-options]
static {
    route 20.20.20.0/27 next-hop 10.10.1.1;
}
```

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
14.2	We recommend that you deactivate and reactivate the service set in such scenarios in Junos OS Release 14.2 and earlier.

CHAPTER 6

Static Destination NAT

IN THIS CHAPTER

- Static Destination NAT | 156

Static Destination NAT

IN THIS SECTION

- Configuring Static Destination Address Translation in IPv4 Networks | 156

Configuring Static Destination Address Translation in IPv4 Networks

To use destination address translation, the size of the pool address space must be greater than or equal to the destination address space. You must specify a name for the `destination-pool` statement, which can contain multiple addresses, ranges, or prefixes, as long as the number of NAT addresses in the pool is larger than the number of destination addresses in the `from` statement.

To configure destination address translation in IPv4 networks:

1. In configuration mode, go to the `[edit services]` hierarchy level.

```
[edit]
user@host# edit services
```

2. Configure the service set and the NAT rule.

```
[edit services]
user@host# set service-set service-set-name nat-rules rule-name
```

In the following example, the name of the service set is **s1** and the name of the NAT rule is **rule-dnat44**.

```
[edit services]
user@host# set service-set s1 nat-rules rule-dnat44
```

3. Go to the `[interface-service]` hierarchy level of the service set.

```
[edit services]
user@host# edit service-set s1 interface-service
```

4. Configure the service interface.

```
 [edit services service-set s1  interface-service]
user@host# set service-interface service-interface-name
```

In the following example, the name of the service interface is **ms-0/1/0**.

> **NOTE**: If the service interface is not present in the router, or the specified interface is not functional, the following command can result in an error.

```
 [edit services service-set s1  interface-service]
user@host# set service-interface ms-0/1/0
```

5. Go to the `[edit services nat]` hierarchy level. Issue the following command from the top of the services hierarchy, or use the **top** keyword.

```
 [edit services service-set s1]
user@host# top editservices nat
```

6. Configure the NAT pool with an address.

```
 [edit services nat]
user@host# set pool pool-name address address
```

In the following example, **dest-pool** is used as the pool name and **4.1.1.2** as the address.

```
user@host# set pool dest-pool address 4.1.1.2
```

7. Configure the rule, match direction, term, and destination address.

```
[edit services nat]
user@host# set rule rule-name match-direction match-direction term term-name from
destination-address  address
```

In the following example, the name of the rule is **rule-dnat44**, the match direction is **input**, the name of the term is **t1**, and the address is **20.20.20.20**.

```
[edit services nat]
user@host# set rule rule-dnat44 match-direction  input term t1 from destination-address
20.20.20.20
```

8. Go to the `[edit services nat rule rule-dnat44 term t1]` hierarchy level.

```
[edit services nat]
user@host# edit rule rule-dnat44 term t1
```

9. Configure the destination pool and the translation type.

```
[edit services nat rule rule-dnat44 term t1]
user@host# set then translated destination-pool dest-pool-name translation-type translation-
type
```

In the following example, the destination pool name is **dest-pool**, and the translation type is **dnat-44**.

```
[edit services nat rule rule-dnat44 term t1]
user@host# set then translated destination-pool  dest-pool translation-type dnat-44
```

10. Go to the `[edit services adaptive-services-pics]` hierarchy level. In the following command, the **top** keyword ensures that the command is run from the top of the hierarchy.

```
[edit services nat rule rule-dnat44 term t1]
user@host# top edit services adaptive-services-pics
```

11. Configure the trace options.

```
[edit services adaptive-services-pics]
user@host# set traceoptions flag tracing parameter
```

In the following example, the tracing parameter is configured as **all**.

```
[edit services adaptive-services-pics]
user@host# set traceoptions flag all
```

12. Verify the configuration by using the `show` command at the `[edit services]` hierarchy level.

```
[edit services]
user@host# show
service-set s1 {
    nat-rules rule-dnat44;
    interface-service {
        service-interface ms-0/1/0;
    }
}
 nat {
    pool dest-pool {
        address 4.1.1.2/32;
    }
    rule rule-dnat44 {
        match-direction input;
        term t1 {
            from {
                destination-address {
                    20.20.20.20/32;
                }
            }
            then {
                translated {
                    destination-pool dest-pool;
                    translation-type {
                        dnat-44;
                    }
                }
            }
        }
    }
}
  adaptive-services-pics {
    traceoptions {
        flag all;
    }
  }
```

The following example configures the translation type as **dnat-44**.

```
[edit services]
user@host# show
service-set s1 {
    nat-rules rule-dnat44;
    interface-service {
        service-interface ms-0/1/0;
    }
}
 nat {
    pool dest-pool {
        address 4.1.1.2/32;
    }
    rule rule-dnat44 {
        match-direction input;
        term t1 {
            from {
                destination-address {
                    20.20.20.20/32;
                }
            }
            then {
                translated {
                    destination-pool dest-pool;
                    translation-type {
                        dnat-44;
                    }
                }
            }
        }
    }
}
  adaptive-services-pics {
    traceoptions {
        flag all;
    }
  }
```

In the following configuration, **term1** configures source address translation for traffic from any private address to any public address. The translation is applied for all services. **term2** performs destination address translation for Hypertext Transfer Protocol (HTTP) traffic from any public address to the server's virtual IP address. The virtual server IP address is translated to an internal IP address.

```
[edit services nat]
rule my-nat-rule {
    match-direction input;
    term my-term1 {
        from {
            source-address private;
            destination-address public;
        }
        then {
            translated {
                source-pool my-pool; # pick address from a pool
                translation-type napt-44; # dynamic NAT with port translation
            }
        }
    }
    }
    rule my-nat-rule2 {
    match-direction input;
    term my-term2 {
        from {
            destination-address 192.168.137.3; # my server's virtual address
            application http;
        }
        then {
            translated {
                destination-pool nat-pool-name;
                translation-type dnat-44; # static destination NAT
            }
        }
    }
}
}
```

The following configuration performs NAT using the destination prefix **20.20.10.0/32** without defining a pool.

```
[edit services nat]
rule src-nat {
    match-direction input;
    term t1 {
        from {
            destination-address 10.10.10.10/32;
            then {
                translation-type dnat44;
                destination-prefix 20.20.10.0/24;
            }
        }
    }
}
```

SEE ALSO

Configuring Source and Destination Addresses Network Address Translation Overview | 92

CHAPTER 7

Network Address Port Translation

IN THIS CHAPTER

- Network Address Port Translation | 163

Network Address Port Translation

IN THIS SECTION

- Configuring Address Pools for Network Address Port Translation (NAPT) Overview | 163
- Configuring NAPT in IPv4 Networks | 170
- Configuring NAPT in IPv6 Networks | 176
- Example: Configuring NAT with Port Translation | 179
- Example: NAPT Configuration on the MS-MPC With an Interface Service Set | 183

Configuring Address Pools for Network Address Port Translation (NAPT) Overview

IN THIS SECTION

- Round-Robin Allocation for NAPT | 164
- Sequential Allocation for NAPT | 165
- Preserve Parity and Preserve Range for NAPT | 166
- Address Pooling and Endpoint Independent Mapping for NAPT | 166
- Secured Port Block Allocation for NAPT | 168
- Comparison of NAPT Implementation Methods | 169

With Network Address Port Translation (NAPT), you can configure up to 32 address ranges with up to 65,536 addresses each.

The `port` statement specifies port assignment for the translated addresses. To configure automatic assignment of ports, include the `port automatic` statement at the `[edit services nat pool` *`nat-pool-name`*`]` hierarchy level. By default, sequential allocation of ports occurs.

Starting with Junos OS Release 14.2, you can include the `sequential` option with the `port automatic` statement at the `[edit services nat pool` *`nat-pool-name`*`]` hierarchy level for sequenced allocation of ports from the specified range. To configure a specific range of port numbers, include the `port range low` *`minimum-value`* `high` *`maximum-value`* statement at the `[edit services nat pool` *`nat-pool-name`*`]` hierarchy level.

> **NOTE**: When 99% of the total available ports in pool for napt-44 , no new flows are allowed on that NAT pool.
>
> Starting with Junos OS Release 14.2, the `auto` option is hidden and is deprecated, and is only maintained for backward compatibility. It might be removed completely in a future software release.

The Junos OS provides several alternatives for allocating ports:

Round-Robin Allocation for NAPT

To configure round-robin allocation for NAT pools, include the **address-allocation round-robin** *configuration statement* at the `[edit services nat pool` *`pool-name`*`]` hierarchy level. When you use round-robin allocation, one port is allocated from each address in a range before repeating the process for each address in the next range. After ports have been allocated for all addresses in the last range, the allocation process wraps around and allocates the next unused port for addresses in the first range.

- The first connection is allocated to the address:port 100.0.0.1:3333.
- The second connection is allocated to the address:port 100.0.0.2:3333.
- The third connection is allocated to the address:port 100.0.0.3:3333.
- The fourth connection is allocated to the address:port 100.0.0.4:3333.
- The fifth connection is allocated to the address:port 100.0.0.5:3333.
- The sixth connection is allocated to the address:port 100.0.0.6:3333.
- The seventh connection is allocated to the address:port 100.0.0.7:3333.
- The eighth connection is allocated to the address:port 100.0.0.8:3333.
- The ninth connection is allocated to the address:port 100.0.0.9:3333.

- The tenth connection is allocated to the address:port 100.0.0.10:3333.
- The eleventh connection is allocated to the address:port 100.0.0.11:3333.
- The twelfth connection is allocated to the address:port 100.0.0.12:3333.
- Wraparound occurs and the thirteenth connection is allocated to the address:port 100.0.0.1:3334.

Sequential Allocation for NAPT

With sequential allocation, the next available address in the NAT pool is selected only when all the ports available from an address are exhausted.

Sequential Allocation can be configured only for the MS-DPC and the MS-100, MS-400, and MS-500 MultiServices PICS. The MS-MPC and MS-MIC cards use only the round-robin allocation approach.

NOTE:

- This legacy implementation provides backward compatibility and is no longer a recommended approach.

The NAT pool called **napt** in the following configuration example uses the sequential implementation:

```
pool napt {
    address-range low 100.0.0.1 high 100.0.0.3;
    address-range low 100.0.0.4 high 100.0.0.6;
    address-range low 100.0.0.8 high 100.0.0.10;
    address-range low 100.0.0.12 high 100.0.0.13;
    port {
        range low 3333 high 3334;
    }
}
```

In this example, the ports are allocated starting from the first address in the first address-range, and allocation continues from this address until all available ports have been used. When all available ports have been used, the next address (in the same address-range or in the following address-range) is allocated and all its ports are selected as needed. In the case of the example **napt** pool, the tuple address, port 100.0.0.4:3333, is allocated only when all ports for all the addresses in the first range have been used.

- The first connection is allocated to the address:port 100.0.0.1:3333.
- The second connection is allocated to the address:port 100.0.0.1:3334.

- The third connection is allocated to the address:port 100.0.0.2:3333.
- The fourth connection is allocated to the address:port 100.0.0.2:3334, and so on.

Preserve Parity and Preserve Range for NAPT

Preserve parity and preserve range options are available for NAPT, and are supported on MS-DPCs and MS-100, MS-400, and MS-500 MultiServices PICS. Support for MS-MPCs and MS-MICs starts in Junos OS Release 15.1R1. The following options are available for NAPT:

- Preserving parity—Use the `preserve-parity` command to allocate even ports for packets with even source ports and odd ports for packets with odd source ports.
- Preserving range—Use the `preserve-range` command to allocate ports within a range from 0 to 1023, assuming the original packet contains a source port in the reserved range. This applies to control sessions, not data sessions.

Address Pooling and Endpoint Independent Mapping for NAPT

IN THIS SECTION

- Address Pooling | **166**
- Endpoint Independent Mapping and Endpoint Independent Filtering | **167**

Address Pooling

Address pooling, or address pooling paired (APP) ensures assignment of the same external IP address for all sessions originating from the same internal host. You can use this feature when assigning external IP addresses from a pool. This option does not affect port utilization

Address pooling solves the problems of an application opening multiple connections. For example, when Session Initiation Protocol (SIP) client sends Real-Time Transport Protocol (RTP) and Real-Time Control Protocol (RTCP) packets, the SIP generally server requires that they come from the same IP address, even if they have been subject to NAT. If RTP and RTCP IP addresses are different, the receiving endpoint might drop packets. Any point-to-point (P2P) protocol that negotiates ports (assuming address stability) benefits from address pooling paired.

The following are use cases for address pooling:

- A site that offers instant messaging services requires that chat and their control sessions come from the same public source address. When the user signs on to chat, a control session authenticates the user. A different session begins when the user starts a chat session. If the chat session originates from a source address that is different from the authentication session, the instant messaging server rejects the chat session, because it originates from an unauthorized address.

- Certain websites such as online banking sites require that all connections from a given host come from the same IP address.

NOTE: Starting with Junos OS Release 14.1, when you deactivate a service-set that contains address pooling paired (APP) for that service-set, messages are displayed on the PIC console and the mappings are cleared for that service-set. These messages are triggered when the deletion of a service-set commences and again generated when the deletion of the service-set is completed. The following sample messages are displayed when deletion starts and ends:

- Nov 15 08:33:13.974 LOG: Critical] SVC-SET ss1 (iid 5) deactivate/delete: NAT Mappings and flows deletion initiated

- Nov 15 08:33:14.674 LOG: Critical] SVC-SET ss1 (iid 5) deactivate/delete: NAT Mappings and flows deletion completed

In a scaled environment that contains a large number of APP in a service set, a heavy volume of messages is generated and this process takes some amount of time. We recommend that you wait until the console messages indicating the completion of deletion of the service set are completed before you reactivate the service-set again.

Endpoint Independent Mapping and Endpoint Independent Filtering

Endpoint independent mapping (EIM) ensures the assignment of the same external address *and* port for all connections from a given host if they use the same internal port. This means if they come from a different source port, you are free to assign a different external address.

EIM and APP differ as follows:

- APP ensures assigning the same external IP address.

- EIM provides a stable external IP address and port (for a period of time) to which external hosts can connect. Endpoint independent filtering (EIF) controls which external hosts can connect to an internal host.

NOTE: Starting with Junos OS Release 14.1, when you deactivate a service-set that contains endpoint independent mapping (EIM) mapping for that service-set, messages are displayed on the PIC console and the mappings are cleared for that service-set. These messages are triggered when the deletion of a service-set commences and again generated when the deletion of the service-set is completed. The following sample messages are displayed when deletion starts and ends:

- Nov 15 08:33:13.974 LOG: Critical] SVC-SET ss1 (iid 5) deactivate/delete: NAT Mappings and flows deletion initiated
- Nov 15 08:33:14.674 LOG: Critical] SVC-SET ss1 (iid 5) deactivate/delete: NAT Mappings and flows deletion completed

In a scaled environment that contains a large number of EIM mappings in a service set, a heavy volume of messages is generated and this process takes some amount of time. We recommend that you wait until the console messages indicating the completion of deletion of the service set are completed before you reactivate the service-set again.

Secured Port Block Allocation for NAPT

IN THIS SECTION

- Secured Port Block Allocation for NAPT | **168**
- Interim Logging for Port Block Allocation | **169**

Port block allocation is supported on MX series routers with MS-DPCs and on M Series routers with MS-100, MS-400, and MS-500 MultiServices PICS. Port block allocation is supported on MX series routers with MS-MPCs and MS-MICs starting in Junos OS release 14.2R2.

Carriers track subscribers using the IP address (RADIUS or DHCP) log. If they use NAPT, an IP address is shared by multiple subscribers, and the carrier must track the IP address and port, which are part of the NAT log. Because ports are used and reused at a very high rate, tracking subscribers using the log becomes difficult due to the large number of messages, which are difficult to archive and correlate. By enabling the allocation of ports in blocks, port block allocation can significantly reduce the number of logs, making it easier to track subscribers.

Secured Port Block Allocation for NAPT

Secured port block allocation can be used for translation types `napt-44` and `stateful-nat64`.

When allocating blocks of ports, the most recently allocated block is the current active block. New requests for NAT ports are served from the active block. Ports are allocated randomly from the current active block.

When you configure secured port block allocation, you can specify the following:

- `block-size`
- `max-blocks-per-address`
- `active-block-timeout`

Interim Logging for Port Block Allocation

With port block allocation we generate one syslog log per set of ports allocated for a subscriber. These logs are UDP based and can be lost in the network, particularly for long-running flows. Interim logging triggers re-sending the above logs at a configured interval for active blocks that have traffic on at least one of the ports of the block.

Interim logging is activated by including the `pba-interim-logging-interval` statement under `services-options` for sp- interfaces.

SEE ALSO

Configuring Secured Port Block Allocation | 267

Configuring NAT Session Logs | 347

Secured Port Block Allocation for NAPT44 and NAT64 Overview | 264

Comparison of NAPT Implementation Methods

Table 1 provides a feature comparison of available NAPT implementation methods.

Table 10: Comparison of NAPT Implementation Methods

Feature/Function	Dynamic Port Allocation	Secured Port Block Allocation	Deterministic Port Block Allocation
Users per IP	High	Medium	Low
Security Risk	Low	Medium	Medium
Log Utilization	High	Low	None (no logs necessary)
Security Risk Reduction	Random allocation	**active-block-timeout** feature	n/a

Table 10: Comparison of NAPT Implementation Methods *(Continued)*

Feature/Function	Dynamic Port Allocation	Secured Port Block Allocation	Deterministic Port Block Allocation
Increasing Users per IP	n/a	Configure multiples of smaller port blocks to maximize users/ public IP	Algorithm-based port allocation

Configuring NAPT in IPv4 Networks

Network Address Port Translation (NAPT) is a method by which many network addresses and their TCP/UDP ports are translated into a single network address and its TCP/UDP ports. This translation can be configured in both IPv4 and IPv6 networks. This section describes the steps for configuring NAPT in IPv4 networks.

To configure NAPT, you must configure a rule at the `[edit services nat]` hierarchy level for dynamically translating the source IPv4 addresses.

To configure the NAPT in IPv4 networks:

1. In configuration mode, go to the `[edit services]` hierarchy level.

   ```
    [edit]
   user@host# edit services
   ```

2. Configure the service set and NAT rule.

   ```
    [edit services]
   user@host# set service-set service-set-name nat-rules rule-name
   ```

   In the following example, the name of the service set is **s1** and the name of the NAT rule is **rule-napt-44**.

   ```
    [edit services]
   user@host# set service-set s1 nat-rules rule-napt-44
   ```

3. Go to the `[interface-service]` hierarchy level of the service set.

```
 [edit services]
user@host# edit service-set s1 interface-service
```

4. Configure the service interface.

```
 [edit services service-set s1 interface service]
user@host# set service-interface service-interface-name
```

In the following example, the name of the service interface is **ms-0/1/0**.

> **NOTE**: If the service interface is not present in the router, or the specified interface is not functional, the following command can result in an error.

```
 [edit services service-set s1 interface service]
user@host# set service-interface ms-0/1/0
```

5. Go to the `[edit services nat]` hierarchy level. Issue the command from the top of the services hierarchy, or use the **top** keyword.

```
[edit services service-set s1 interface service]
user@host# top edit services nat
```

6. Configure the NAT pool with an address.

```
 [edit services nat]
user@host# set pool pool-name address address
```

In the following example, the name of the pool is **napt-pool** and the address is **10.10.10.0**.

```
 [edit services nat]
user@host# set pool napt-pool address 10.10.10.0
```

7. Configure the port.

```
 [edit services nat]
user@host# set pool pool-name port port-type
```

In the following example, the port type is selected as **sequential** or `auto`.

```
 [edit services nat]
user@host# set pool napt-pool port automatic
```

NOTE: Starting in Junos OS Release 14.2, the `sequential` option is introduced to enable you to configure sequential allocation of ports. The `sequential` and `random-allocation` options available with the `port automatic` statement at the `[edit services nat pool` *`nat-pool-name`*`]` hierarchy level are mutually exclusive. You can include the `sequential` option for sequential allocation and the `random-allocation` option for random delegation of ports. By default, sequential allocation of ports takes place if you include only the `port automatic` statement at the `[edit services nat pool` *`nat-pool- name`*`]` hierarchy level. The `auto` option is hidden and is deprecated in Junos OS Release 14.2 and later, and is only maintained for backward compatibility. It might be removed completely in a future software release.

8. Configure the rule and the match direction.

```
 [edit services nat]
user@host# set rule rule-name match-direction match-direction
```

In the following example, the name of the rule is **rule-napt-44** and the match direction is **input**.

```
 [edit services nat]
user@host# set rule rule-napt-44 match-direction input
```

9. Configure the term, the action for the translated traffic, and the translation type.

```
 [edit services nat]
user@host# set rule rule-name term term-name then translated translated-action translation-
type napt-44
```

In the following example, the name of the term is **t1**, the action for the translated traffic is **translated**, the name of the source pool is **napt-pool**, and the translation type is **napt-44**.

```
 [edit services nat]
user@host# set rule rule-napt-44 match-direction  input term t1 then translated source-pool
napt-pool translation-type napt-44
```

10. Go to the `[edit services adaptive-services-pics]` hierarchy level. In the command, the **top** keyword ensures that the command is run from the top of the hierarchy.

```
[edit services nat]
user@host# top edit services adaptive-services-pics
```

11. Configure the trace options.

```
[edit services adaptive-services-pics]
user@host# set traceoptions flag tracing parameter
```

In the following example, the tracing parameter is configured as **all**.

```
[edit services adaptive-services-pics]
user@host# set traceoptions flag all
```

12. Verify the configuration by using the `show` command at the `[edit services]` hierarchy level.

```
[edit services]
user@host# show
service-set s1 {
    nat-rules rule-napt-44;
    interface-service {
        service-interface ms-0/1/0;
    }
}
nat {
    pool napt-pool {
        address 10.10.10.0/32;
        port {
            automatic;
        }
    }
    rule rule-napt-44 {
        match-direction input;
        term t1 {
            then {
                translated {
                    source-pool napt-pool;
                    translation-type {
                        napt-44;
                    }
                }
            }
        }
    }
}
adaptive-services-pics {
    traceoptions {
        flag all;
    }
}
```

The following example configures the translation type as **napt-44**.

```
[edit services]
    user@host# show
    service-set s1 {
    nat-rules rule-napt-44;
    interface-service {
        service-interface ms-0/1/0;
    }
}
nat {
    pool napt-pool {
        address 10.10.10.0/32;
        port {
            automatic auto;
        }
    }
    rule rule-napt-44 {
        match-direction input;
        term t1 {
            then {
                translated {
                    source-pool napt-pool;
                    translation-type {
                        napt-44;
                    }
                }
            }
        }
    }
}
adaptive-services-pics {
    traceoptions {
        flag all;
    }
}
```

Dynamic Address Translation to a Small Pool with Fallback to NAT

The following configuration shows dynamic address translation from a large prefix to a small pool, translating a **/24** subnet to a pool of 10 addresses. When the addresses in the source pool (**src-pool**) are exhausted, NAT is provided by the NAPT overload pool (**pat-pool**).

```
[edit services nat]
pool src-pool {
    address-range low 192.16.2.1 high 192.16.2.10;
}
pool pat-pool {
    address-range low 192.16.2.11 high 192.16.2.12;
    port automatic auto;
    rule myrule {
    match-direction input;
    term myterm {
        from {
            source-address 10.150.1.0/24;
        }
        then {
            translated {
                source-pool src-pool;
                overload-pool pat-pool;
                translation-type napt-44;
            }
        }
    }
}
```

Dynamic Address Translation with Small Pool

The following configuration shows dynamic address translation from a large prefix to a small pool, translating a **/24** subnet to a pool of 10 addresses. Sessions from the first 10 host sessions are assigned an address from the pool on a first-come, first-served basis, and any additional requests are rejected. Each host with an assigned NAT can participate in multiple sessions.

```
[edit services nat]
pool my-pool {
    address-range low 10.10.10.1 high 10.10.10.10;
}
rule src-nat {
    match-direction input;
    term t1 {
        from {
            source-address 192.168.1.0/24;
        }
        then {
            translated {
                translation-type dynamic-nat44;
                source-pool my-pool;
            }
        }
    }
}
```

Configuring NAPT in IPv6 Networks

Network Address Port Translation (NAPT) is a method by which many network addresses and their TCP/UDP ports are translated into a single network address and its TCP/UDP ports. This translation can be configured in both IPv4 and IPv6 networks. This section describes the steps for configuring NAPT in IPv6 networks. Configuring NAPT in IPv6 networks is not supported if you are using MS-MPCs or MS-MICs. For information about configuring NAPT in IPv4 networks, see "Configuring NAPT in IPv4 Networks" on page 170 .

To configure NAPT, you must configure a rule at the `[edit services nat]` hierarchy level for dynamically translating the source IPv6 addresses.

To configure NAPT in IPv6 networks:

1. In configuration mode, go to the `[edit services nat]` hierarchy level.

   ```
    [edit]
   user@host# edit services nat
   ```

2. Define the pool of IPv6 source addresses that must be used for dynamic translation. For NAPT, also specify port numbers when configuring the source pool.

   ```
   [edit services nat]
   user@host# set pool pool name address IPv6 source addresses
   user@host# set pool pool name port source ports
   ```

   For example:

   ```
    [edit services nat]
   user@host#  set pool IPV6-NAPT-Pool address 2002::1/96
   user@host#  set pool IPV6-NAPT-Pool port automatic sequential
   ```

3. Define a NAT rule for translating the source addresses. To do this, set the `match-direction` statement of the rule as `input`. In addition, define a term that uses `napt-66` as the translation type for translating the addresses of the pool defined in the previous step. Note that the `napt-66` translation type is supported only on the MS-DPC, MS-100, MS-400, and MS-500 line cards.

   ```
    [edit services nat]
   user@host# set rule rule name match-direction input
   user@host# set rule rule name term term name then translated source-pool pool name
   user@host# set rule rule name term term name then translated translation-type napt-66
   ```

   For example:

   ```
    [edit services nat]
   user@host# set rule IPV6-NAPT-Rule match-direction input
   user@host# set rule IPV6-NAPT-Rule term t1 then translated source-pool IPV6-NAPT-Pool
   user@host# set rule IPV6-NAPT-Rule term t1 then translated translation-type napt-66
   ```

4. Enter the `up` command to navigate to the `[edit services]` hierarchy level.

```
 [edit services nat]
user@host# up
```

5. Define a service set to specify the services interface that must be used, and reference the NAT rule implemented for NAPT translation.

```
 [edit services]
user@host# set service-set service-set name interface- service service-interface services
interface
user@host# set service-set service-set name nat-rules rule name
```

For example:

```
 [edit services]
user@host# set service-set IPV6-NAPT-ServiceSet interface-service service-interface sp-0/1/0
user@host# set service-set IPV6-NAPT-ServiceSet nat-rules IPV6-NAPT-Rule
```

6. Define the trace options for the adaptive services PIC.

```
 [edit services]
user@host# set adaptive-services-pics traceoptions flag tracing parameter
```

For example:

```
 [edit services]
user@host# set adaptive-services-pics traceoptions flag all
```

The following example configures dynamic source (address and port) translation or NAPT for an IPv6 network.

```
[edit services]
user@host# show
    service-set IPV6-NAPT-ServiceSet {
        nat-rules IPV6-NAPT-Rule;
        interface-service {
            service-interface sp-0/1/0;
        }
    }
    nat {
        pool IPV6-NAPT-Pool {
            address 2002::1/96;
            port automatic sequential;
        }
        rule IPV6-NAPT-Rule {
            match-direction input;
            term term1 {
                then {
                    translated {
                        source-pool IPV6-NAPT-Pool;
                        translation-type {
                            napt-66;
                        }
                    }
                }
            }
        }
    }
    adaptive-services-pics {
        traceoptions {
            flag all;
        }
    }
}
```

Example: Configuring NAT with Port Translation

IN THIS SECTION

- Requirements | 180
- Overview | 180
- Configuring NAT with Port Translation | 180

This example shows how to configure NAT with port translation.

Requirements

This example uses the following hardware and software components:

- An MX Series 5G Universal Routing Platform with a Services DPC or an M Series Multiservice Edge router with a services PIC
- A domain name server (DNS)
- Junos OS Release 11.4 or higher

Overview

This example shows a complete CGN NAT44 configuration and advanced options.

Configuring NAT with Port Translation

IN THIS SECTION

- Procedure | 180

Procedure

Step-by-Step Procedure

To configure the service set:

1. Configure a service set.

```
user@host# edit services service-set ss2
```

2. In configuration mode, go to the `[edit services nat]` hierarchy level.

```
 [edit]
user@host# edit services nat
```

3. Define the pool of source addresses that must be used for dynamic translation. For NAPT, also specify port numbers when configuring the source pool.

```
[edit services nat]
user@host# set pool pool name address source addresses
user@host# set pool pool name port source ports
```

For example:

```
 [edit services nat]
user@host#  set pool NAPT-Pool address 192.168.2.1/24;
user@host#  set pool NAPT-Pool port automatic
```

4. Specify the NAT rule to be used.

```
[edit services service-set ss2]
host# set nat-rules r1
```

5. Define a NAT rule for translating the source addresses. To do this, set the `match-direction` statement of the rule as `input`. In addition, define a term that uses `napt-44` as the translation type for translating the addresses of the pool defined in the previous step.

```
 [edit services nat]
user@host# set rule rule name match-direction input
user@host# set rule rule name term term name from source-address source-address
user@host# set rule rule name term term name then translated source-pool pool name
user@host# set rule rule name term term name then translated translation-type napt-44
```

For example:

```
 [edit services nat]
user@host# set rule r1 match-direction input
user@host# set rule r1 term t1 from source-address 10.10.10.1
user@host# set rule r1 term t1 then translated source-pool NAPT-Pool
user@host# set rule r1 term t1 then translated translation-type napt-44
```

6. Specify the interface service.

```
[edit services service-set ss2]
host# set interface-service service-interface sp-5/0/0
```

Results

```
user@host# show services service-sets sset2

service-set ss2 {
        nat-rules r1;
        interface-service {
            service-interface sp-5/0/0;
        }
    }
    nat {
        pool NAPT-Pool {
            address 192.168.2.1/24;
            port automatic;
        }
        rule r1 {
            match-direction input;
            term t1 {
                from {
                    source-address {
                        10.10.10.1/32;
                    }
                }
                then {
                    translated {
                        source-pool NAPT-Pool;
                        translation-type {
                            napt-44;
                        }
                    }
                }
            }
        }
    }
```

Example: NAPT Configuration on the MS-MPC With an Interface Service Set

IN THIS SECTION

- Requirements | 183
- Overview | 183
- Configuration | 183

This example shows how to configure network address translation with port translation (NAPT) on an MX series router using a MultiServices Modular Port Concentrator (MS-MPC) as a services interface card.

Requirements

This example uses the following hardware and software components:

- MX-series router
- MultiServices Modular Port Concentrator (MS-MPC)
- Junos OS Release 13.2R1 or higher

Overview

A service provider has chosen an MS-MPC as a platform to provide NAT services to accommodate new subscribers.

Configuration

IN THIS SECTION

- CLI Quick Configuration | 184
- Configuring Interfaces | 185
- Configure an Application Set of Acceptable Application Traffic | 185
- Configuring a Stateful Firewall Rule | 186
- Configuring NAT Pool and Rule | 187

- Configuring the Service Set | 188

To configure NAPT44 using the MS-MPC as a services interface card, perform these tasks:

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the [edit] hierarchy level.

```
set interfaces ge-0/2/0 unit 0 family inet address 10.255.248.2/24
set interfaces  xe-1/1/0  unit 0 family inet address 10.255.247.2/24
set interfaces  xe-1/1/0  unit 0 family inet service input service-set sset1
set interfaces  xe-1/1/0  unit 0 family inet service output service-set sset1
set interfaces ms-3/0/0 unit 0 family inet
set applications application-set accept-algs application junos-http
set applications application-set accept-algs application junos-ftp
set applications application-set accept-algs application junos-tftp
set applications application-set accept-algs application junos-telnet
set applications application-set accept-algs application junos-sip
set applications application-set accept-algs application junos-rtcp
set services stateful-firewall rule sf-rule1 match-direction input-output
set services stateful-firewall rule sf-rule1 term sf-term1 from source-address  10.255.247.0/24
set services stateful-firewall rule sf-rule1 term sf-term1 from application-sets accept-algs
set services stateful-firewall rule sf-rule1 term sf-term1 then accept
set services nat pool napt-pool address 1.1.1.0/24
set services nat pool napt-pool port automatic
* nat rule for napt
set services nat rule nat-rule1 match-direction input
set services nat rule nat-rule1 term nat-term1 from source-address 10.255.247.0/24
set services nat rule nat-rule1 term nat-term1 from application-sets accept-algs
set services nat rule nat-rule1 term nat-term1 then translated source-pool napt-pool
set services nat rule nat-rule1 term nat-term1 then translated translation-type napt-44
* nat rule for basic nat
set services service-set sset1 stateful-firewall-rules sf-rule1
set services service-set sset1 nat-rules nat-rule1
set services service-set sset1 interface-service  service-interface ms-3/0/0
```

Configuring Interfaces

Step-by-Step Procedure

Configure the interfaces required for NAT processing. You will need the following interfaces:

- A customer-facing interface that handles traffic from and to the customer.
- An internet-facing interface.
- A services interface that provides NAT and stateful firewall services to the customer-facing interface

1. Configure the interface for the customer-facing interface.

```
user@host# edit
[edit ]
user@host# set interfaces  xe-1/1/0  unit 0 family inet address 10.255.247.2/24
user@host# set interfaces  xe-1/1/0  unit 0 family inet service input service-set sset1
user@host# set interfaces  xe-1/1/0  unit 0 family inet service output service-set sset1
```

2. Configure the interface for the Internet-facing interface.

```
[edit ]
set interfaces ge-0/2/0 unit 0 family inet address 10.255.248.2/24
```

3. Configure the interface for the service set that will connect services to the customer-facing interface. In our example, the interface resides on an MS-MPC.

```
[edit ]
user@host# set interfaces ms-3/0/0 unit 0 family inet
```

Configure an Application Set of Acceptable Application Traffic

Step-by-Step Procedure

Identify the acceptable applications for incoming traffic.

1. Specify an application set that contains acceptable incoming application traffic.

```
user@host# set applications application-set accept-algs application junos-http
user@host# set applications application-set accept-algs application junos-ftp
user@host# set applications application-set accept-algs application junos-tftp
user@host# set applications application-set accept-algs application junos-telnet
user@host# set applications application-set accept-algs application junos-sip
user@host# set applications application-set accept-algs application junos-rtcp
```

Results

```
user@host#edit services applications application-set accept-algs
user@host#show
application junos-http;
application junos-ftp;
application junos-tftp;
application junos-telnet;
application junos-sip;
application junos-
```

Configuring a Stateful Firewall Rule

Step-by-Step Procedure

Configure a stateful firewall rule that will accept all incoming traffic.

1. Specify firewall matching for all input and output

```
user@hos#t set services stateful-firewall rule sf-rule1 match-direction input-output
```

2. Identify source-address and acceptable application traffic from the customer-facing interface.

```
user@host# set services stateful-firewall rule sf-rule1 term sf-term1 from source-address
10.255.247.0/24
user@host# set services stateful-firewall rule sf-rule1 term sf-term1 from application-sets
accept-algs
user@host# set services stateful-firewall rule sf-rule1 term sf-term1 then accept
```

Results

```
user@host# edit services stateful-firewall
user@host# show
rule sf-rule1 {
    match-direction input-output;
    term sf-term1 {
        from {
            source-address {
                10.255.247.0/24;
            }
            application-sets accept-algs;
        }
        then {
            accept;
        }
    }
}
```

Configuring NAT Pool and Rule

Step-by-Step Procedure

Configure a NAT pool and rule for address translation with automatic port assignment.

1. Configure the NAT pool with automatic port assignment.

   ```
   user@host# set services nat pool napt-pool address 1.1.1.0/24
   user@host# set services nat pool napt-pool port automatic auto
   ```

2. Configure a NAT rule that applies translation type `napt-44` using the defined NAT pool.

   ```
   user@host# set services nat rule nat-rule1 term nat-term1 from application-sets accept-algs
   user@host# set services nat rule nat-rule1 term nat-term1 then translated source-pool napt-
   pool
   user@host# set services nat rule nat-rule1 term nat-term1 then translated translation-type
   napt-44
   ```

Results

```
user@host#edit services nat
user@host#show

pool napt-pool {
    address 1.1.1.0/24;
    port {
        automatic;
    }
}
rule nat-rule1 {
    match-direction input;
    term nat-term1 {
        from {
            source-address {
                10.255.247.0/24;
            }
            application-sets accept-algs;
        }
        then {
            translated {
                source-pool napt-pool;
                translation-type {
                    napt-44;
                }
            }
        }
    }
}
```

Configuring the Service Set

Step-by-Step Procedure

Configure an interface type service set.

1. Specify the NAT and stateful firewall rules that apply to customer traffic.

```
user@host set services service-set sset1 stateful-firewall-rules sf-rule1
user@host set services service-set sset1 nat-rules bat-rule1
```

2. Specify the services interface that applies the rules to customer traffic.

```
set services service-set sset1 interface-service  service-interface ms-3/0/0
```

Results

```
user@host# edit services service-set sset1
user@host# show
set services service-set sset1 stateful-firewall-rules sf-rule1
set services service-set sset1 nat-rules nat-rule1
set services service-set sset1 interface-service service-interface ms-3/0/0
```

SEE ALSO

Junos Address Aware Network Addressing Overview | 53

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
14.2	Starting with Junos OS Release 14.2, you can include the `sequential` option with the `port automatic` statement at the `[edit services nat pool` *`nat-pool-name`*`]` hierarchy level for sequenced allocation of ports from the specified range.
14.2	Starting with Junos OS Release 14.2, the `auto` option is hidden and is deprecated, and is only maintained for backward compatibility.
14.2	Starting in Junos OS Release 14.2, the `sequential` option is introduced to enable you to configure sequential allocation of ports.

14.1	Starting with Junos OS Release 14.1, when you deactivate a service-set that contains address pooling paired (APP) for that service-set, messages are displayed on the PIC console and the mappings are cleared for that service-set.
14.1	Starting with Junos OS Release 14.1, when you deactivate a service-set that contains endpoint independent mapping (EIM) mapping for that service-set, messages are displayed on the PIC console and the mappings are cleared for that service-set.

CHAPTER 8

Deterministic NAT

IN THIS CHAPTER

- Deterministic NAT | 191

Deterministic NAT

IN THIS SECTION

- Deterministic NAPT Overview | 191
- Configuring Deterministic NAPT | 197

Deterministic NAPT Overview

IN THIS SECTION

- Benefits of Deterministic NAPT | 192
- Understanding Deterministic NAPT Algorithms | 192
- Deterministic NAPT Restrictions | 195

You can configure deterministic NAPT44 to ensure that the original source IPv4 address and port always map to the same post-NAT IPv4 address and port range, and that the reverse mapping of a given translated external IPv4 address and port are always mapped to the same internal IPv4 address. You can configure deterministic NAPT64 to ensure that the original source IPv6 address and port always map to the same post-NAT IPv4 address and port range, and that the reverse mapping of a given translated external IPv4 address and port are always mapped to the same internal IPv6 address. Deterministic NAPT uses an algorithm-based allocation of blocks of destination ports.

Deterministic NAPT44 is supported on MX series routers with MS-DPCs and on M Series routers with MS-100, MS-400, and MS-500 MultiServices PICS. Deterministic NAPT 44 is supported for MS-MPCs and MS-MICs starting in Junos OS release 17.3R1, in Junos OS release 14.2R7 and later 14.2 releases, and in Junos OS release 15.1R3 and later 15.1 releases. Starting in Junos OS Release 17.4R1, deterministic NAPT64 is supported on the MS-MPC and MS-MIC.

If the source address in the `from` clause of a deterministic NAPT rule does not have a prefix of /32, the network and broadcast addresses in the source address range are not translated unless you configure `include-boundary-addresses`.

For detailed information on how to configure deterministic NAPT, see "Configuring Deterministic NAPT" on page 197 .

Benefits of Deterministic NAPT

- Eliminates the need for address translation logging because an IP address is always mapped to the same external IP address and port range, and the reverse mapping of a given translated external IP address and port are always mapped to the same internal IP address.

Understanding Deterministic NAPT Algorithms

The effectiveness of your implementation of deterministic NAPT depends on your analysis of your subscriber requirements. The block size you provide indicates how many ports will be made available for each incoming subscriber address from the range in the `from` clause specified in the applicable NAT rule. The allocation algorithm computes an offset value to determine the outgoing IP address and port. A reverse algorithm is used to derive the originating subscriber address.

> **NOTE**: In order to track subscribers without using logs, an ISP must use a reverse algorithm to derive a subscriber (source) addresses from a translated address.

The following variables are used in forward calculation (private subscriber IP address to public IP address) and reverse calculation (public IP address to private subscriber IP address):

- Pr_Prefix—Any pre-NAT IPv4 subscriber address.
- Pr_Port—Any pre-NAT protocol port.
- Block_Size—Number of ports configured to be available for each Pr_Prefix.

  If `block-size` is configured as zero, the method for computing the block size is computed as follows:

block-size = int(64512/ceil[(Nr_Addr_PR_Prefix/Nr_Addr_PU_Prefix)])

where 64512 is the maximum available port range per public IP address.

- Base_PR_Prefix—First usable pre-NAT IPv4 subscriber address in a `from` clause of the NAT rule.
- Base_PU_Prefix—First usable post-NAT IPv4 subscriber address configured in the NAT pool.
- Pu_Port_Range_Start—First usable post-NAT port. This is 1024.
- Pr_Offset—The offset of the pre-NAT IP address that is being translated from the first usable pre-NAT IPv4 subscriber address in a `from` clause of the NAT rule. PR_Offset = Pr_Prefix – Base_Pr_Prefix.
- PR_Port_Offset—Offset of the pre-NAT IP address multiplied by the block size. PR_Port_Offset = Pr_Offset * Block_Size.
- Pu_Prefix—Post-NAT address for a given Pr_Prefix.
- Pu_Start_Port—Post-NAT start port for a flow from a given Pr_Prefix
- Pu_Actual_Port—Post-NAT port seen on a reverse flow.
- Nr_Addr_PR_Prefix — Number of usable pre-NAT IPv4 subscriber addresses in a `from` clause clause of the NAT rule.
- Nr_Addr_PU_Prefix — Number of usable post-NAT IPv4 addresses configured in the NAT pool.
- Rounded_Port_Range_Per_IP — Number of ports available for each post-NAT IP address. Rounded_Port_Range_Per_IP = ceil[(Nr_Addr_PR_Prefix/Nr_Addr_PU_Prefix)] * Block_Size.
- Pu_Offset—Offset of the post-NAT IP address from the first usable post-NAT address. Pu_Offset = Pu_Prefix – Base_Pu_Prefix.
- Pu_Port_Offset— Offset of the post-NAT port from 1024 added to the product of the offset of the post-NAT IP address and the number of ports available for each post-NAT IP address. Pu_Port_Offset = (Pu_Offset * Rounded_Port_Range_Per_IP) + (Pu_Actual_Port – Pu_Port_Range_Start).

Algorithm Usage–Assume the following configuration:

```
services {
    nat {
        pool src-pool {
            address-range low 32.32.32.1 high 32.32.32.254;
            port {
                automatic {
                    random-allocation;
                }
                deterministic-block-allocation {
                    block-size 249;
                }
            }
        }
rule det-nat {
match-direction input;
    term t1 {
        from {
            source-address {
                10.1.0.0/16;
            }
        }
        then {
            translated {
                source-pool src-pool;
                translation-type {
                    deterministic-napt44;
                }
            }
        }
    }
```

Forward Translation

1. Pr_Offset = Pr_Prefix – Base_Pr_Prefix

2. Pr_Port_Offset = Pr_Offset * Block_Size

3. Rounded_Port_Range_Per_IP = ceil[(Nr_Addr_PR_Prefix/Nr_Addr_PU_Prefix)] * Block_Size

4. Pu_Prefix = Base_Public_Prefix + floor(Pr_Port_Offset / Rounded_Port_Range_Per_IP)

5. Pu_Start_Port = Pu_Port_Range_Start + (Pr_Port_Offset % Rounded_Port_Range_Per_IP)

Using the sample configuration and assuming a subscriber flow sourced from 10.1.1.250:5000:

1. Pr_Offset = 10.1.1.250 – 10.1.0.1 = 505

2. Pr_Port_Offset = 505 * 249 = 125,745

3. Rounded_Port_Range_Per_IP = ceil[(65, 533/254)] * 249 = 259 * 249 = 64,491

4. Pu_Prefix = 32.32.32.1 + floor(125,745 /64,491) = 32.32.32.1 +1 =32.32.32.2

5. Pu_Start_Port = 1,024 + (125,745 % 64,491) = 62278

   - 10.1.1.250 is translated to 32.32.32.2.
   - The starting port is 62278. There are 249 ports available to the subscriber based on the configured block size. The available port range spans ports 62278 through 62526 (inclusive).
   - The specific flow 10.1.1.250:5000 randomly assigns any of the ports in its range because random allocation was specified.

Reverse Translation

1. Pu_Offset = Pu_Prefix – Base_Pu_Prefix
2. Pu_Port_Offset = (Pu_Offset * Rounded_Port_Range_Per_IP) + (Pu_Actual_Port – Pu_Port_Range_Start)
3. Subscriber_IP = Base_Pr_Prefix + floor(Pu_Port_Offset / Block_Size)

The reverse translation is determined as follows. Assume a flow returning to 32.32.32.2:62278.

1. Pu_Offset = 32.32.32.2 – 32.32.32.1 = 1
2. Pu_Port_Offset = (1 * 64,491) + (62,280 - 1024) = 125,747
3. Subscriber_IP = 10.1.0.1 + floor(125,747 / 249) = 10.1.0.1 + 505 = 10.1.1.250

NOTE: In reverse translation, only the original private IP address can be derived, and not the original port in use. This is sufficiently granular for law enforcement requirements.

When you have configured deterministic NAPT, you can use the `show services nat deterministic-nat internal-host` and `show services nat deterministic-nat nat-port-block` commands to show forward and reverse mapping. However, mappings will change if you reconfigure your deterministic port block allocation block size or the `from` clause for your NAT rule. In order to provide historical information on mappings, we recommend that you write scripts that can show specific mappings for prior configurations.

Deterministic NAPT Restrictions

When you configure deterministic NAPT, you must be aware of the following restrictions. Violation of any restriction results in a commit error. The restrictions and their error messages are shown in Table 11 on page 196 .

Table 11: Deterministic NAPT Commit Constraints

Restriction	Error Message
The total number of deterministic NAT blocks must be greater than or equal to the `from` clause addresses configured. This means that the Rounded_Port_Range_Per_IP value must be less than or equal to 64,512.	Number of addresses and port blocks combination in the NAT pool is less than number of addresses in 'from' clause
IPv6 addresses should not be used in deterministic NAT pool/ from clause.	Invalid IP address in pool p1 with translation type deterministic-napt44 OR There is already a range configured with v4 address range
The `from` clause addresses should be same if the same deterministic NAT pool is used across multiple terms/rules. Only one `from` clause address/range should be specified if the same deterministic NAT pool is used across multiple terms/rules.	With translation-type deterministic-napt44, same 'from' address/range should be configured if pool is shared by multiple rules or terms
The `from` clause must have at least one source address.	With translation-type deterministic-napt44, at least one non-except 'from' address/ range should be configured. error: configuration check-out failed
There should not be address overlap between `except` entries in the `from` clause addresses.	overlapping address, in the 'from' clause between 'except' entries
Addresses in a NAT pool used for deterministic NAPT should not overlap with the addresses in any other NAT pool.	NAT pool det-nat-pool1 overlaps with det-nat-pool used by service set sset_det-nat error: configuration check-out failed
A deterministic NAT pool cannot be used with other translation types. In addition, a deterministic NAT pool cannot be used in both deterministic NAPT44 and deterministic NAPT64 NAT rules.	Deterministic NAT pool cannot be used with other translation-types

Table 11: Deterministic NAPT Commit Constraints *(Continued)*

Restriction	Error Message
Deterministic NAPT44 must use a source pool with deterministic-port-block-allocation configuration.	Deterministic NAPT44 must use a source pool with deterministic-port-block-allocation configuration
If `address-allocation round-robin` is configured, a commit results in display of a warning indicating that this technique is not needed with translation-type deterministic-napt44 and is ignored.	Address allocation round-robin is not needed with translation-type deterministic-napt44
The total number of IP addresses assigned to a deterministic NAT pool should be less than or equal to 2^{24} (16777216).	Number of addresses in pool with deterministic-napt44 translation are limited to at most 16777216(2^24)

Configuring Deterministic NAPT

IN THIS SECTION

- Configuring the NAT Pool for Deterministic NAPT | 197
- Configuring the NAT Rule for Deterministic NAPT | 199
- Configuring the Service Set for Deterministic NAT | 200

Deterministic NAPT44 is supported on MX series routers with MS-DPCs and on M Series routers with MS-100, MS-400, and MS-500 MultiServices PICS. Deterministic NAPT44 is supported for MS-MPCs and MS-MICs starting in Junos OS release 17.3R1, in Junos OS release 14.2R7 and later 14.2 releases, and in Junos OS release 15.1R3 and later 15.1 releases. Starting in Junos OS Release 17.4R1, deterministic NAPT64 is supported on the MS-MPC and MS-MIC.

To configure deterministic NAPT, perform the following:

Configuring the NAT Pool for Deterministic NAPT

To configure the NAT pool for deterministic NAPT:

1. At the [edit services nat pool *poolname*] hierarchy level, create a pool.

   ```
   user@host# edit services nat pool poolname
   ```

2. Define the range of addresses to be translated, specifying the upper and lower limits of the range or an address prefix that describes the range.

   ```
   [edit services nat pool pba-pool1]
   user@host# set address-range low address high address
   ```

   Or

   ```
   user@host# set address address-prefix
   ```

3. To configure automatic port assignment, specify either sequential or random allocation.

   ```
   [edit services nat pool pba-pool1]
   user@host# set port automatic (sequential | random-allocation)
   ```

   NOTE: Starting in Junos OS Release 14.2R1, the sequential option is introduced to enable you to configure sequential allocation of ports. The sequential and random-allocation options available with the port automatic statement at the [edit services nat pool *nat-pool-name*] hierarchy level are mutually exclusive. You can include the sequential option for sequential allocation and the random-allocation option for random delegation of ports. By default, sequential allocation of ports takes place if you include only the port automatic statement at the [edit services nat pool *nat-pool- name*] hierarchy level.

   For releases earlier than Junos OS Release 14.2R1, configure automatic sequential port assignment by using the auto option at the [edit services nat pool *nat-pool-name* port automatic] hierarchy level.

4. To configure a range of ports to assign, specify the low and high values for the port. If you do not configure automatic port assignment, you must configure a range of ports.

NOTE: If you specify a range of ports to assign, the `automatic` statement is ignored.

```
[edit services nat pool pba-pool1]
user@host# set port range low minimum-value high maximum-value
```

5. Configure deterministic port block allocation. Specify **block-size** or accept the default value of 512.

   You can also specify `include-boundary-addresses` if you want the lowest and highest addresses (the network and broadcast addresses) in the source address range of a NAT rule to be translated when the NAT pool is used. If the source address has a prefix of /32, the lowest and highest address are automatically translated.

   ```
   [edit services nat pool pba-pool1]
   user@host# set port deterministic-port-block-allocation block-size block-size include-
   boundary-addresses
   ```

   For example:

   ```
   [edit services nat pool pba-pool1]
   user@host# set port deterministic-port-block-allocation block-size 256
   ```

   NOTE: In order for `deterministic-port-block-allocation` configuration changes to take effect, you must reboot the services PIC whenever you change any of the following `nat pool` options:

   - `address` or `address-range`
   - `port range`
   - `port deterministic-port-block-allocation block-size`

SEE ALSO

Network Address Translation Configuration Overview | 92

Configuring the NAT Rule for Deterministic NAPT

To configure the NAT rule for deterministic NAPT:

1. Configure the NAT rule name.

```
 [edit services nat]
user@host# set rule rule-name
```

2. Configure the NAT rule match direction as input.

```
 [edit services nat]
user@host# set rule rule-name match-direction input
```

3. Specify the addresses that are translated by the NAT rule.

   To specify one address:

```
 [edit services nat]
user@host# set rule rule-name term term-name from source-addressaddress
```

   To specify a range of addresses:

```
 [edit services nat]
user@host# set rule rule-name term term-name from source-address-range low minimum-value high
maximum-value
```

4. Specify the NAT pool that contains the addresses for translated traffic.

```
 [edit services nat]
user@host# set rule rule-name term term-name then translated source-pool nat-pool-name
```

5. Configure the translation type as deterministic NAPT44 or deterministic NAPT64.

```
 [edit services nat]
user@host# set rule rule-name term term-name then translation-type (deterministic-napt44 |
deterministic-napt64)
```

Configuring the Service Set for Deterministic NAT

To configure the service set for deterministic NAPT:

1. Define the service set.

   ```
    [edit services]
   user@host# edit service-set service-set-name
   ```

2. Configure either an interface service, which requires a single service interface, or a next-hop service, which requires an inside and outside service interface.

   ```
    [edit services service-set service-set-name]
   user@host# set interface-service service-interface interface-name
   ```

   or

   ```
    [edit services service-set service-set-name]
   user@host# set next-hop-service inside-service-interface interface-name outside-service-
   interface interface-name
   ```

3. Specify the NAT rules or ruleset to be used with the service set.

   ```
    [edit services service-set service-set-name]
   user@host# set nat-rules rule-name
   ```

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
17.4R1	Starting in Junos OS Release 17.4R1, deterministic NAPT64 is supported on the MS-MPC and MS-MIC.
17.4R1	Starting in Junos OS Release 17.4R1, deterministic NAPT64 is supported on the MS-MPC and MS-MIC.
17.3R1	Deterministic NAPT 44 is supported for MS-MPCs and MS-MICs starting in Junos OS release 17.3R1
17.3R1	Deterministic NAPT44 is supported for MS-MPCs and MS-MICs starting in Junos OS release 17.3R1
14.2R1	Starting in Junos OS Release 14.2R1, the `sequential` option is introduced to enable you to configure sequential allocation of ports.

CHAPTER 9

NAT Protocol Translation

IN THIS CHAPTER

- NAT Protocol Translation | 202
- Example: Configuring the DNS ALG Application on MX-SPC3 service card | 233

NAT Protocol Translation

IN THIS SECTION

- Configuring NAT-PT | 202
- Example: Configuring NAT-PT | 212

Configuring NAT-PT

IN THIS SECTION

- Configuring the DNS ALG Application | 203
- Configuring the NAT Pool and NAT Rule | 203
- Configuring the Service Set for NAT | 208
- Configuring Trace Options | 210

To configure the translation type as `basic-nat-pt`, you must configure the DNS ALG application, the NAT pools and rules, a service set with a service interface, and trace options. Configuring NAT-PT is not supported if you are using MS-MPCs or MS-MICs. This topic includes the following tasks:

Configuring the DNS ALG Application

To configure the DNS ALG application:

1. In configuration mode, go to the `[edit applications]` hierarchy level.

   ```
    [edit]
   user@host# edit applications
   ```

2. Configure the ALG to which the DNS traffic is destined at the `[edit applications]` hierarchy level. Define the application name and specify the application protocol to use in match conditions in the first NAT rule or term.

   ```
    [edit applications]
   user@host# set application application-name application-protocol application-protocol
   ```

   In the following example, the application name is **dns-alg** and application protocol is **dns**.

   ```
    [edit applications]
   user@host# set application dns-alg application-protocol dns
   ```

3. Verify the configuration by using the `show` command at the `[edit applications]` hierarchy level.

   ```
   [edit applications]
   user@host# show
   application dns-alg {
       application-protocol dns;
   }
   ```

Configuring the NAT Pool and NAT Rule

To configure the NAT pool and NAT rule:

1. In configuration mode, go to the `[edit services nat]` hierarchy level.

   ```
    [edit]
   user@host# edit services nat
   ```

2. Configure the NAT pool and its address.

   ```
    [edit services nat]
   user@host# set pool pool-name address address
   ```

   In the following example, the name of the NAT pool is **p1** and the address is **10.10.10.2/32**.

   ```
    [edit services nat]
   user@host# set pool p1 address 10.10.10.2/32
   ```

3. Configure the source pool and its address.

   ```
    [edit services nat]
   user@host# set pool source-pool-name address address
   ```

   In the following example, the name of the source pool is **src_pool0** and the source pool address is **20.1.1.1/32**.

   ```
    [edit services nat]
   user@host# set pool src_pool0 address 20.1.1.1/32
   ```

4. Configure the destination pool and its address.

   ```
    [edit services nat]
   user@host# set pool destination-pool-name address address
   ```

   In the following example, the name of the destination pool is **dst_pool0** and the destination pool address is **50.1.1.2/32**.

   ```
    [edit services nat]
   user@host# set pool dst_pool0 address 50.1.1.2/32
   ```

5. Configure the rule and the match direction.

   ```
    [edit services nat]
   user@host# set rule rule-name match-direction match-direction
   ```

In the following example, the rule name is **rule-basic-nat-pt** and the match direction is **input**.

```
 [edit services nat]
user@host# set rule basic-nat-pt match-direction input
```

6. Configure the term and the input conditions for the NAT term.

```
 [edit services nat]
user@host# set rule rule-basic-nat-pt term term from from
```

In the following example, the term is **t1** and the input conditions are **source-address 2000::2/128**, **destination-address 4000::2/128**, and **applications dns_alg**.

```
 [edit services nat]
user@host# set rule rule-basic-nat-pt term t1 from source-address 2000::2/128
 [edit services nat]
user@host# set rule rule-basic-nat-pt term t1 from destination-address 4000::2/128
 [edit services nat]
user@host# set rule rule-basic-nat-pt term t1 from applications dns_alg
```

7. Configure the NAT term action and the properties of the translated traffic.

```
 [edit services nat]
user@host# set rule rule-basic-nat-pt term t1 then term-action translated-property
```

In the following example, the term action is **translated** and the properties of the translated traffic are **source-pool src_pool0**, **destination-pool dst_pool0**, and **dns-alg-prefix 2001:db8:10::0/96**.

```
 [edit services nat]
user@host# set rule rule-basic-nat-pt term t1 then translated source-pool src_pool0
 [edit services nat]
user@host# set rule rule-basic-nat-pt term t1 then translated destination-pool dst_pool0
 [edit services nat]
user@host# set rule rule-basic-nat-pt term t1 then translated dns-alg-prefix
2001:db8:10::0/96
```

8. Configure the translation type.

```
 [edit services nat]
user@host# set rule rule-basic-nat-pt term t1 then translated translation-type translation-
type
```

In the following example, the translation type is **basic-nat-pt**.

```
 [edit services nat]
user@host# set rule rule-basic-nat-pt term t1 then translated translation-type basic-nat-pt
```

9. Configure another term and the input conditions for the NAT term.

```
 [edit services nat]
user@host# set rule rule-basic-nat-pt term term-name from from
```

In the following example, the term name is **t2** and the input conditions are **source-address 2000::2/128** and **destination-address 2001:db8:10::0/96**.

```
 [edit services nat]
user@host# set rule rule-basic-nat-pt term t2 from source-address 2000::2/128
 [edit services nat]
user@host# set rule rule-basic-nat-pt term t2 from destination-address 2001:db8:10::0/96
```

10. Configure the NAT term action and the property of the translated traffic.

```
 [edit services nat]
user@host# set rule rule-basic-nat-pt term t2 then term-action translated-property
```

In the following example, the term action is **translated** and the property of the translated traffic is **source-prefix 19.19.19.1/32**.

```
 [edit services nat]
user@host# set rule rule-basic-nat-pt term t2 then translated source-prefix 19.19.19.1/32
```

11. Configure the translation type.

```
 [edit services nat]
user@host# set rule rule-basic-nat-pt term t2 then translated translation-type translation-
type
```

In the following example, the translation type is **basic-nat-pt**.

```
 [edit services nat]
user@host# set rule rule-basic-nat-pt term t2 then translated translation-type basic-nat-pt
```

12. Verify the configuration by using the `show` command at the `[edit services nat]` hierarchy level.

```
[edit services nat]
user@host# show
pool p1 {
    address 10.10.10.2/32;
}
pool src_pool0 {
    address 20.1.1.1/32;
}
pool dst_pool0 {
    address 50.1.1.2/32;
}
rule rule-basic-nat-pt {
    match-direction input;
    term t1 {
        from {
            source-address {
                2000::2/128;
            }
            destination-address {
                4000::2/128;
            }
            applications dns_alg;
        }
        then {
            translated {
                source-pool src_pool0;
                destination-pool dst_pool0;
                dns-alg-prefix 2001:db8:10::0/96;
                translation-type {
                    basic-nat-pt;
                }
            }
        }
    }
    term t2 {
        from {
            source-address {
                2000::2/128;
            }
            destination-address {
                2001:db8:10::0/96;
            }
        }
        then {
            translated {
                source-prefix 19.19.19.1/32;
                translation-type {
                    basic-nat-pt;
                }
            }
        }
    }
}
```

Configuring the Service Set for NAT

To configure the service set for NAT:

1. In configuration mode, go to the `[edit services]` hierarchy level.

```
 [edit]
user@host# edit services
```

2. Configure the service set.

```
 [edit services]
user@host# edit service-set service-set-name
```

In the following example, the name of the service set is **ss_dns**.

```
 [edit services]
user@host# edit service-set ss_dns
```

3. Configure the service set with NAT rules.

```
 [edit services service-set ss_dns]
user@host# set nat-rules rule-name
```

In the following example, the rule name is **rule-basic-nat-pt**.

```
 [edit services service-set ss_dns]
user@host# set nat-rules rule-basic-nat-pt
```

4. Configure the service interface.

```
 [edit services service-set ss_dns]
user@host# set interface-service service-interface service-interface-name
```

In the following example, the name of service interface is **sp-1/2/0**.

```
 [edit services service-set ss_dns]
user@host# set interface-service service-interface sp-1/2/0
```

5. Verify the configuration by using the `show services` command from the `[edit]` hierarchy level.

```
[edit]
user@host# show services
    service-set ss_dns {
        nat-rules rule-basic-nat-pt;
        interface-service {
            service-interface sp-1/2/0;
        }
    }
```

Configuring Trace Options

To configure the trace options:

1. In configuration mode, go to the `[edit services adaptive-services-pics]` hierarchy level.

   ```
   [edit]
   user@host# edit services adaptive-services-pics
   ```

2. Configure the trace options.

   ```
   [edit services adaptive-services-pics]
   user@host# set traceoptions flag tracing parameter
   ```

   In the following example, the tracing parameter is **all**.

   ```
   [edit services adaptive-services-pics]
   user@host# set traceoptions flag all
   ```

3. Verify the configuration by using the `show` command at the `[edit services]` hierarchy level.

   ```
   [edit services]
   user@host# show
   adaptive-services-pics {
       traceoptions {
           flag all;
       }
   }
   ```

The following example configures the translation type as **basic-nat-pt**.

```
[edit]
user@host# show services
service-set ss_dns {
    nat-rules rule-basic-nat-pt;
    interface-service {
        service-interface sp-1/2/0;
    }
}
```

```
nat {
    pool p1 {
        address 10.10.10.2/32;
    }
    pool src_pool0 {
        address 20.1.1.1/32;
    }
    pool dst_pool0 {
        address 50.1.1.2/32;
    }
    rule rule-basic-nat-pt {
        match-direction input;
        term t1 {
            from {
                source-address {
                    2000::2/128;
                }
                destination-address {
                    4000::2/128;
                }
                applications dns_alg;
            }
            then {
                translated {
                    source-pool src_pool0;
                    destination-pool dst_pool0;
                    dns-alg-prefix 2001:db8:10::0/96;
                    translation-type {
                        basic-nat-pt;
                    }
                }
            }
        }
        term t2 {
            from {
                source-address {
                    2000::2/128;
                }
                destination-address {
                    2001:db8:10::0/96;
                }
            }
            then {
                translated {
                    source-prefix 19.19.19.1/32;
                    translation-type {
                        basic-nat-pt;
                    }
                }
            }
        }
    }
}
adaptive-services-pics {
    traceoptions {
        flag all;
    }
}
```

Example: Configuring NAT-PT

IN THIS SECTION

- Requirements | 213
- Overview and Topology | 213
- Configuration of NAT-PT with DNS ALGs | 215

A Domain Name System application-level gateway (DNS ALG) is used with Network Address Translation-Protocol Translation (NAT-PT) to facilitate name-to-address mapping. You can configure the DNS ALG to map addresses returned in the DNS response to an IPv6 address. Configuring NAT-PT is not supported if you are using MS-MPCs or MS-MICs.

When you configure NAT-PT with DNS ALG support, you must configure two NAT rules or one rule with two terms. In this example, you configure two rules. The first NAT rule ensures that the DNS query and response packets are translated correctly. For this rule to work, you must configure a DNS ALG application and reference it in the rule. The second rule is required to ensure that NAT sessions are destined to the address mapped by the DNS ALG.

Then, you must configure a service set, and then apply the service set to the interfaces.

This example describes how to configure NAT-PT with DNS ALG:

Requirements

This example uses the following hardware and software components:

- Junos OS Release 11.2
- A multiservices interface (**ms-**)

Overview and Topology

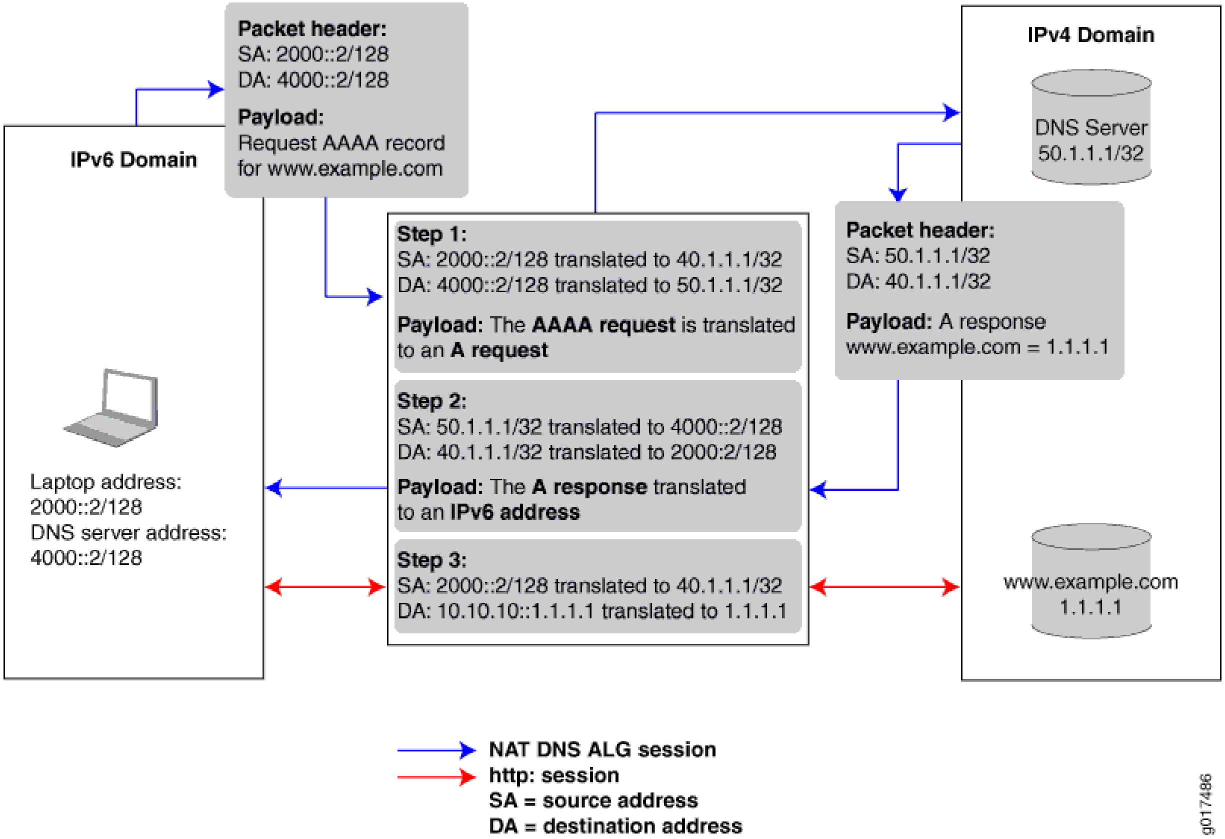
IN THIS SECTION

● Topology | **214**

The following scenario shows the process of NAT-PT with DNS ALG when a laptop in an IPv6-only domain requests access to a server in an IPv4-only domain.

Topology

Figure 10: Configuring DNS ALGs with NAT-PT Network Topology



The Juniper Networks router in the center of the illustration performs address translation in two steps. When the laptop requests a session with the **www.example.com** server that is in an IPv4-only domain, the Juniper Networks router performs the following:

- Translates the IPv6 laptop and DNS server addresses into IPv4 addresses.

- Translates the AAAA request from the laptop into an A request so that the DNS server can provide the IPv4 address.

When the DNS server responds with the A request, the Juniper Networks router performs the following:

- Translates the IPv4 DNS server address back into an IPv6 address.

- Translates the A request back into a AAAA request so that the laptop now has the 96-bit IPv6 address of the **www.example.com** server.

After the laptop receives the IPv6 version of the **www.example.com** server address, the laptop initiates a second session using the 96-bit IPv6 address to access that server. The Juniper Networks router performs the following:

- Translates the laptop IPv4 address directly into its IPv4 address.
- Translates the 96-bit IPv6 **www.example.com** server address into its IPv4 address.

Configuration of NAT-PT with DNS ALGs

IN THIS SECTION

- Configuring the Application-Level Gateway | **215**
- Configuring the NAT Pools | **217**
- Configuring the DNS Server Session: First NAT Rule | **218**
- Configuring the HTTP Session: Second NAT Rule | **223**
- Configuring the Service Set | **226**
- Configuring the Stateful Firewall Rule | **228**
- Configuring Interfaces | **230**

To configure NAT-PT with DNS ALG , perform the following tasks:

Configuring the Application-Level Gateway

Step-by-Step Procedure

Configure the DNS application as the ALG to which the DNS traffic is destined. The DNS application protocol closes the DNS flow as soon as the DNS response is received. When you configure the DNS application protocol, you must specify the UDP protocol as the network protocol to match in the application definition.

To configure the DNS application:

1. In configuration mode, go to the `[edit applications]` hierarchy level.

```
user@host# edit applications
```

2. Define the application name and specify the application protocol to use in match conditions in the first NAT rule.

```
 [edit applications]
user@host#  set application application-name application-protocol protocol-name
```

For example:

```
 [edit applications]
user@host#  set application dns_alg application-protocol dns
```

3. Specify the protocol to match, in this case UDP.

```
 [edit applications]
user@host#  set application application-name protocol  type
```

For example:

```
[edit applications]
user@host# set application dns_alg protocol udp
```

4. Define the UDP destination port for additional packet matching, in this case the domain port.

```
 [edit applications]
user@host#  set application application-name destination-port  value
```

For example:

```
[edit applications]
user@host# set application dns_alg destination-port 53
```

Results

```
[edit applications]
user@host# show
application dns_alg {
    application-protocol dns;
    protocol udp;
    destination-port 53;
}
```

Configuring the NAT Pools

Step-by-Step Procedure

In this configuration, you configure two pools that define the addresses (or prefixes) used for NAT. These pools define the IPv4 addresses that are translated into IPv6 addresses. The first pool includes the IPv4 address of the source. The second pool defines the IPv4 address of the DNS server. To configure NAT pools:

1. In configuration mode, go to the `[edit services nat]` hierarchy level.

   ```
   user@host# edit services nat
   ```

2. Specify the name of the first pool and the IPv4 source address (laptop).

   ```
   [edit services nat]
   user@host# set pool nat-pool-name address ip-prefix
   ```

   For example:

   ```
   [edit services nat]
   user@host# set pool pool1 address 40.1.1.1/32
   ```

3. Specify the name of the second pool and the IPv4 address of the DNS server.

   ```
   [edit services nat]
   user@host# set pool nat-pool-name address ip-prefix
   ```

   For example:

   ```
   [edit services nat]
   user@host# set pool pool2 address 50.1.1.1/32
   ```

Results

The following sample output shows the configuration of NAT pools.

```
[edit services nat]
user@host# show
pool pool1 {
    address 40.1.1.1/32;
}
pool pool2 {
    address 50.1.1.1/32;
}
```

Configuring the DNS Server Session: First NAT Rule

Step-by-Step Procedure

The first NAT rule is applied to DNS traffic going to the DNS server. This rule ensures that the DNS query and response packets are translated correctly. For this rule to work, you must configure a DNS ALG application and reference it in the rule. The DNS application was configured in "Configuring NAT-PT" on page 202 . In addition, you must specify the direction in which traffic is matched, the source address of the laptop, the destination address of the DNS server, and the actions to take when the match conditions are met.

To configure the first NAT rule:

1. In configuration mode, go to the `[edit services nat]` hierarchy level.

   ```
   user@host# edit services nat
   ```

2. Specify the name of the NAT rule.

   ```
   [edit services nat]
   user@host# edit rule rule-name
   ```

For example:

```
[edit services nat]
user@host# edit rule rule1
```

3. Specify the name of the NAT term.

   ```
   [edit services nat rule rule-name]
    user@host# edit term term-name
   ```

   For example:

   ```
   [edit services nat rule rule1]
    user@host# edit term term1
   ```

4. Define the match conditions for this rule.

   - Specify the IPv6 source address of the device (laptop) attempting to access an IPv4 address.

     ```
     [edit services nat rule rule-name term term-name]
     user@host# set from source-address source-address
     ```

     For example:

     ```
     [edit services nat rule rule1 term term1]
     user@host# set from source-address 2000::2/128
     ```

   - Specify the IPv6 destination address of the DNS server.

     ```
     [edit services nat rule rule-name term term-name]
     user@host# set from destination-address prefix
     ```

     For example:

     ```
     [edit services nat rule rule1 term term1]
      user@host# set from destination-address 4000::2/128
     ```

- Reference the DNS application to which the DNS traffic destined for port 53 is applied.

  ```
  [edit services nat rule rule1 term term1]
  user@host# set from applications application-name
  ```

  In this example, the application name configured in the *Configuring the DNS Application* step is **dns_alg**:

  ```
  [edit services nat rule rule1 term term1]
  user@host# set from applications dns_alg
  ```

5. Define the actions to take when the match conditions are met. The source and destination pools you configured in "Configuring the NAT Pools" on page 217 are applied here.

   - Apply the NAT pool configured for source translation.

     ```
     [edit services nat rule rule-name term term-name]
     user@host# set then translated source-pool nat-pool-name
     ```

     For example:

     ```
     [edit services nat rule rule1 term term1]
      user@host# set then translated source-pool pool1
     ```

   - Apply the NAT pool configured for destination translation.

     ```
     [edit services nat rule rule-name term term-name]
     user@host# set then translated destination-pool nat-pool-name
     ```

     For example:

     ```
     [edit services nat rule rule1 term term1]
      user@host# set then translated source-pool pool2
     ```

6. Define the DNS ALG 96-bit prefix for IPv4-to-IPv6 address mapping.

   ```
   [edit services nat rule rule-name term term-name]
   user@host# set then translated dns-alg-prefix dns-alg-prefix
   ```

   For example:

   ```
   [edit services nat rule rule1 term term1]
    user@host# set then translated dns-alg-prefix 10:10:10::0/96
   ```

7. Specify the type of NAT used for source and destination traffic.

   ```
   [edit services nat rule rule-name term term-name]
   user@host# set then translated translation-type basic-nat-pt
   ```

   For example:

   ```
   [edit services nat rule rule1 term term1]
   user@host# set then translated translation-type basic-nat-pt
   ```

   NOTE: In this example, since NAT is achieved using address-only translation, the **basic-nat-pt** translation type is used. To achieve NAT using address and port translation (NAPT), use the **napt-pt** translation type.

8. Specify the direction in which to match traffic that meets the rule conditions.

   ```
   [edit services nat rule rule-name]
   user@host# set match-direction (input | output)
   ```

   For example:

   ```
   [edit services nat rule rule1]
   user@host# set match-direction input
   ```

9. Configure system logging to record information from the services interface to the /var/log directory.

```
[edit services nat rule rule-name term term-name]
user@host# set then syslog
```

For example:

```
 [edit services nat rule rule1 term term1]
user@host# set then syslog
```

Results

The following sample output shows the configuration of the first NAT rule that goes to the DNS server.

```
[edit services nat]
user@host# show
rule rule1 {
    match-direction input;
    term term1 {
        from {
            source-address {
                2000::2/128;
            }
            destination-address {
                4000::2/128;
            }
            applications dns_alg;
        }
        then {
            translated {
                source-pool pool1;
                destination-pool pool2;
                dns-alg-prefix 10:10:10::0/96;
                translation-type {
                    basic-nat-pt;

                }
            }
            syslog;
        }
    }
}
```

Configuring the HTTP Session: Second NAT Rule

Step-by-Step Procedure

The second NAT rule is applied to destination traffic going to the IPv4 server (**www.example.com**). This rule ensures that NAT sessions are destined to the address mapped by the DNS ALG. For this rule to work, you must configure the DNS ALG address map that correlates the DNS query or response processing done by the first rule with the actual data sessions processed by the second rule. In addition, you must specify the direction in which traffic is matched: the IPv4 address for the IPv6 source address (laptop), the 96-bit prefix to prepend to the IPv4 destination address (www.example.com), and the translation type.

To configure the second NAT rule:

1. In configuration mode, go to the following hierarchy level.

   ```
   user@host# edit services nat
   ```

2. Specify the name of the NAT rule and term.

   ```
   [edit services nat]
   user@host# edit rule rule-name term term-name
   ```

   For example:

   ```
   [edit services nat]
   user@host# edit rule rule2 term term1
   ```

3. Define the match conditions for this rule:

   - Specify the IPv6 address of the device attempting to access the IPv4 server.

     ```
     [edit services nat rule rule-name term term-name]
     user@host# set from source-address source-address
     ```

For example:

```
[edit services nat rule rule2 term term1]
user@host# set from source-address 2000::2/128
```

- Specify the 96-bit IPv6 prefix to prepend to the IPv4 server address.

```
[edit services nat rule rule-name term term-name]
user@host# set from destination-address prefix
```

For example:

```
[edit services nat rule rule2 term term1]
 user@host# set from destination-address 10:10:10::c0a8:108/128
```

4. Define the actions to take when the match conditions are met.

- Specify the prefix for the translation of the IPv6 source address.

```
[edit services nat rule rule-name term term-name]
user@host# set then translated source-prefix source-prefix
```

For example:

```
[edit services nat rule rule2 term term1]
 user@host# set then translated source-prefix 19.19.19.1/32
```

5. Specify the type of NAT used for source and destination traffic.

```
[edit services nat rule rule-name term term-name]
user@host# set then translated translation-type basic-nat-pt
```

For example:

```
[edit services nat rule rule2 term term1]
user@host# set then translated translation-type basic-nat-pt
```

NOTE: In this example, since NAT is achieved using address-only translation, the **basic-nat-pt** translation type is used. To achieve NAT using address and port translation (NAPT), you must use the **napt-pt** translation type.

6. Specify the direction in which to match traffic that meets the conditions in the rule.

```
[edit services nat rule rule-name]
user@host# set match-direction (input | output)
```

For example:

```
[edit services nat rule rule2]
user@host# set match-direction input
```

Results

The following sample output shows the configuration of the second NAT rule.

```
[edit services nat]
user@host# show
rule rule2 {
    match-direction input;
    term term1 {
        from {
            source-address {
                2000::2/128;
            }
            destination-address {
                10:10:10::c0a8:108/128;
            }
        }
        then {
            translated {
                source-prefix 19.19.19.1/32;
                translation-type {
                    basic-nat-pt;
                }
            }
        }
    }
}
```

Configuring the Service Set

Step-by-Step Procedure

This service set is an interface service set used as an action modifier across the entire services (**ms-**) interface. Stateful firewall and NAT rule sets are applied to traffic processed by the services interface.

To configure the service set:

1. In configuration mode, go to the `[edit services]` hierarchy level.

   ```
   user@host# edit services
   ```

2. Define a service set.

   ```
   [edit services]
   user@host# edit service-set service-set-name
   ```

   For example:

   ```
   [edit services]
   user@host# edit service-set ss
   ```

3. Specify properties that control how system log messages are generated for the service set.

   ```
   [edit services service-set ss]
   user@host# set syslog host local services severity-level
   ```

   The example below includes all severity levels.

   ```
   [edit services service-set ss]
   user@host# set syslog host local services any
   ```

4. Specify the stateful firewall rule included in this service set.

   ```
   [edit services service-set ss]
   user@host# set stateful-firewall-rules rule1 severity-level
   ```

   The example below references the stateful firewall rule defined in "Configuring the Stateful Firewall Rule" on page 228 .

   ```
   [edit services service-set ss]
   user@host# set stateful-firewall-rules rule1
   ```

5. Define the NAT rules included in this service set.

   ```
   [edit services service-set ss]
   user@host# set nat-rules rule-name
   ```

   The example below references the two rules defined in this configuration example.

   ```
   [edit services service-set ss
   user@host# set nat-rules rule1
   user@host# set nat-rules rule2
   ```

6. Configure an adaptive services interface on which the service is to be performed.

   ```
   [edit services service-set ss]
   user@host# set interface-service service-interface interface-name
   ```

   For example:

   ```
   [edit services service-set ss
   user@host# interface-service service-interface ms-2/0/0
   ```

   Only the device name is needed, because the router software manages logical unit numbers automatically. The services interface must be an adaptive services interface for which you have configured **unit 0 family inet** at the `[edit interfaces *interface-name*]` hierarchy level in "Configuring Interfaces" on page 230 .

Results

The following sample output shows the configuration of the service set.

```
[edit services]
user@host# show
service-set ss {
    syslog {
        host local {
            services any;
        }
    }
    stateful-firewall-rules rule1;
    nat-rules rule1;
    nat-rules rule2;
    interface-service {
        service-interface ms-2/0/0;
    }
}
```

Configuring the Stateful Firewall Rule

Step-by-Step Procedure

This example uses a stateful firewall to inspect packets for state information derived from past communications and other applications. The NAT-PT router checks the traffic flow matching the direction specified by the rule, in this case both input and output. When a packet is sent to the services (**ms-**) interface, direction information is carried along with it.

To configure the stateful firewall rule:

1. In configuration mode, go to the `[edit services stateful firewall]` hierarchy level.

   ```
   user@host# edit services stateful firewall
   ```

2. Specify the name of the stateful firewall rule.

   ```
    [edit services stateful-firewall]
   user@host# edit rule rule-name
   ```

For example:

```
 [edit services stateful-firewall]
user@host# edit rule rule1
```

3. Specify the direction in which traffic is to be matched.

```
[edit services stateful-firewall rule rule-name]
user@host# set match-direction (input | input-output | output)
```

For example:

```
 [edit services stateful-firewall rule rule1]
user@host# set match-direction input-output
```

4. Specify the name of the stateful firewall term.

```
 [edit services stateful-firewall rule rule-name]
user@host# edit term  term-name
```

For example:

```
 [edit services stateful-firewall rule rule1]
user@host# edit term term1
```

5. Define the terms that make up this rule.

```
 [edit services stateful-firewall rule rule-name term term-name]
user@host# set then accept
```

For example:

```
 [edit services stateful-firewall rule rule1 term term1]
user@host# set then accept
```

Results

The following sample output shows the configuration of the services stateful firewall.

```
[edit services]
user@host# show
stateful-firewall {
    rule rule1 {
        match-direction input-output;
        term term1 {
            then {
                accept;
            }
        }
    }
}
```

Configuring Interfaces

Step-by-Step Procedure

After you have defined the service set, you must apply services to one or more interfaces installed on the router. In this example, you configure one interface on which you apply the service set for input and output traffic. When you apply the service set to an interface, it automatically ensures that packets are directed to the services (**ms-**) interface.

To configure the interfaces:

1. In configuration mode, go to the `[edit interfaces]` hierarchy level.

   ```
   user@host# edit interfaces
   ```

2. Configure the interface on which the service set is applied to automatically ensure that packets are directed to the services (**ms-**) interface.

   - For IPv4 traffic, specify the IPv4 address.

     ```
     [edit interfaces]
     user@host# set ge-1/0/9 unit 0 family inet address 30.1.1.1/24
     ```

- Apply the service set defined in "Configuring Interfaces" on page 230 .

  ```
  [edit interfaces]
  user@host# set ge-1/0/9 unit 0 family inet6 service input service-set ss
  user@host# set ge-1/0/9 unit 0 family inet6 service output service-set ss
  ```

- For IPv6 traffic, specify the IPv6 address.

  ```
  [edit interfaces]
  user@host# set ge-1/0/9 unit 0 family inet6 address 2000::1/64
  ```

3. Specify the interface properties for the services interface that performs the service.

   ```
   [edit interfaces]
   user@host# set ms-2/0/0 services-options syslog host local services any
   user@host# set ms-2/0/0 unit 0 family inet
   user@host# set ms-2/0/0 unit 0 family inet6
   ```

Results

The following sample output shows the configuration of the interfaces for this example.

```
[edit interfaces]
user@host# show

ge-1/0/9 {
    unit 0 {
        family inet {
            address 30.1.1.1/24;
        }
        family inet6 {
            service {
                input {
                    service-set ss;
                }
                output {
```

```
                    service-set ss;
                }
            }
            address 2000::1/64;
        }
    }
}

ms-2/0/0 {
    services-options {
        syslog {
            host local {
                services any;
            }
        }
    }
    unit 0 {
        family inet;
        family inet6;
    }
}
```

SEE ALSO

- Junos Address Aware Network Addressing Overview | **53**
- Configuring NAT-PT | **202**
- Configuring Service Sets to be Applied to Services Interfaces | **10**
- Example: Configuring Layer 3 Services and the Services SDK on Two PICs | **562**
- *dns-alg-prefix*
- *dns-alg-pool*

Example: Configuring the DNS ALG Application on MX-SPC3 service card

SUMMARY

IN THIS SECTION

- Requirements | 233
- Configuration | 233

This example shows how to configure the translation type as basic-nat-pt. You must configure the DNS ALG application, the NAT pools and rules, a service set with a service interface.

Requirements

This example uses the following hardware and software components:

- MX240, MX480, and MX960 with MX-SPC3
- Junos OS Release 21.1R1

Configuration

To configure the DNS ALG application on the MX-SPC3 service card, perform these tasks:

1. Set the application.

   ```
    [edit]
   user@host#  set application application-name application-protocol protocol-name
   ```

2. Configuring service set.

   ```
    [edit]
   user@host# set services service-set ss1 syslog mode event
   ```

   ```
   user@host# set services service-set ss1 syslog mode event
   ```

3. 3. Configure a service set using the NAT rule.

```
 [edit]
user@host# set services service-set ss1 nat-rule-sets src_nat_rule_set1
```

```
user@host# set services service-set ss1 nat-rule-sets dst_nat_rule_set1
```

```
user@host# set services service-set ss1 interface-service service-interface vms-2/0/0.0
```

4. Specify NAT pool and rule information.

```
 [edit]
user@host# set services nat source pool source_pool1 address 100.0.0.0/24
```

```
user@host# set services nat source rule-set src_nat_rule_set1 rule source_nat_rule1 match
source-address 2000::/64
```

```
user@host# set services nat source rule-set src_nat_rule_set1 rule source_nat_rule1 match
destination-address 0.0.0.0/0
```

```
user@host# set services nat source rule-set src_nat_rule_set1 rule source_nat_rule1 match
application dns_alg
```

```
user@host# set services nat source rule-set src_nat_rule_set1 rule source_nat_rule1 then
source-nat pool source_pool1
```

```
user@host# set services nat source rule-set src_nat_rule_set1 rule source_nat_rule1 then
syslog
```

```
user@host# set services nat source rule-set src_nat_rule_set1 match-direction input
```

```
user@host# set services nat destination rule-set dst_nat_rule_set1 rule dst_nat_rule1 match
source-address 2000::/64
```

```
user@host# set services nat destination rule-set dst_nat_rule_set1 rule dst_nat_rule1 match
destination-address 6000::/96
```

```
user@host# set services nat destination rule-set dst_nat_rule_set1 rule dst_nat_rule1 match
application dns_alg
```

```
user@host# set services nat destination rule-set dst_nat_rule_set1 rule dst_nat_rule1 then
```

```
destination-nat destination-prefix 6000::/96
```

```
user@host# set services nat destination rule-set dst_nat_rule_set1 rule dst_nat_rule1 then
syslog
```

```
user@host# set services nat destination rule-set dst_nat_rule_set1 match-direction input
```

5. Configure the interfaces.

```
 [edit]
user@host# set interfaces vms-2/0/0 unit 0 family inet
```

```
user@host# set interfaces vms-2/0/0 unit 0 family inet6
```

Result

```
[edit]
user@host# show services service-set ss1 {
    syslog {
        mode event;
        local-category all;
    }
    nat-rule-sets src_nat_rule_set1;
    nat-rule-sets dst_nat_rule_set1;
    interface-service {
        service-interface vms-2/0/0.0;
    }
}
nat {
    source {
        pool source_pool1 {
            address {
                100.0.0.0/24;
            }
        }
        rule-set src_nat_rule_set1 {
            rule source_nat_rule1 {
                match {
                    source-address 2000::/64;
                    destination-address 0.0.0.0/0;
                    application dns_alg;
                }
                then {
                    source-nat {
                        pool {
                            source_pool1;
                        }
                    }
                    syslog;
                }
            }
            match-direction input;
        }
    }
    destination {
        rule-set dst_nat_rule_set1 {
            rule dst_nat_rule1 {
                match {
                    source-address 2000::/64;
                    destination-address 6000::/96;
                    application dns_alg
                }
                then {
                    destination-nat {
                        destination-prefix 6000::/96;
                    }
                    syslog;
                }
            }
            match-direction input;
        }
    }
}
```

CHAPTER 10

IPv4 Connectivity Across IPv6-Only Network Using 464XLAT

IN THIS CHAPTER

- IPv4 Connectivity Across IPv6-Only Network Using 464XLAT | **238**

IPv4 Connectivity Across IPv6-Only Network Using 464XLAT

IN THIS SECTION

- 464XLAT Overview | **238**
- Configuring 464XLAT Provider-Side Translater for IPv4 Connectivity Across IPv6-Only Network | **240**

464XLAT Overview

IN THIS SECTION

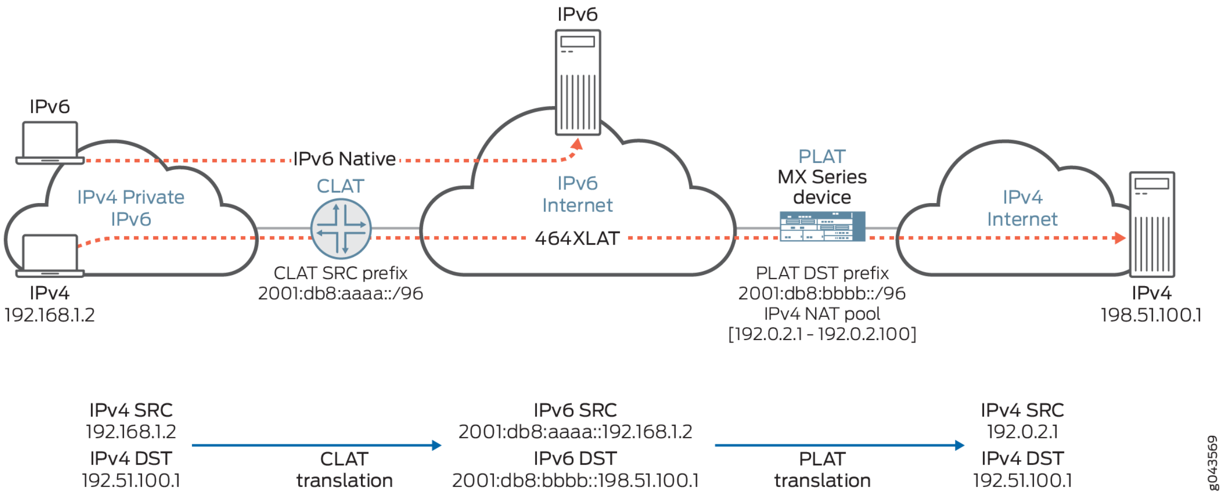
- Benefits of 464XLAT | **240**

Starting in Junos OS Release 17.1R1, you can configure a 464XLAT Provider-Side Tranlator (PLAT). This is supported only on MS-MICs and MS-MPCs. 464XLAT provides a simple and scalable technique for an IPv4 client with a private address to connect to an IPv4 host over an IPv6 network. 464XLAT only suports IPv4 in the client-server model, so it does not support IPv4 peer-to-peer communication or inbound IPv4 connections.

XLAT464 provides the advantages of not having to maintain an IPv4 network for this IPv4 traffic and not having to assign additional public IPv4 addresses.

A customer-side translator (CLAT), which is not a Juniper Networks product, translates the IPv4 packet to IPv6 by embedding the IPv4 source and destination addresses in IPv6 /96 prefixes, and sends the packet over an IPv6 network to the PLAT. The PLAT translates the packet to IPv4, and sends the packet to the IPv4 host over an IPv4 network (see Figure 11 on page 239).

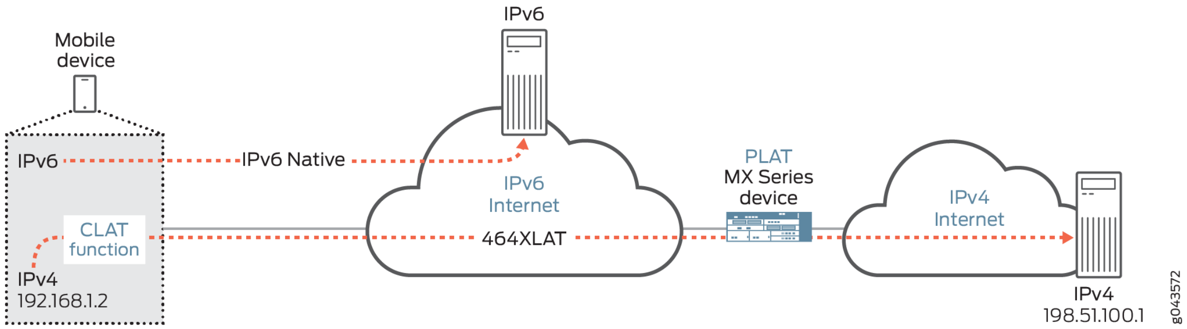
Figure 11: 464XLAT Wireline Flow



The CLAT uses a unique source IPv6 prefix for each end user, and translates the IPv4 source address by embedding it in the IPv6 /96prefix. In Figure 11 on page 239 , the CLAT source IPv6 prefix is 2001:db8:aaaa::/96, and the IPv4 source address 192.168.1.2 is translated to 2001:db8:aaaa::192.168.1.2. The CLAT translates the IPv4 destination address by embedding it in the IPv6 /96 prefix of the PLAT (MX Series router). In Figure 11 on page 239 , the PLAT destination IPv6 prefix is 2001:db8:bbbb::/96, so the CLAT translates the IPv4 destination address 198.51.100.1 to 2001:db8:bbbb::198.51.100.

The CLAT can reside on the end user mobile device in an IPv6-only mobile network, allowing mobile network providers to roll out IPv6 for their users *and* support IPv4-only applications on mobile devices (see Figure 12 on page 240).

Figure 12: 464XLAT Wireless Flow



To configure the PLAT on the MX Series router, you create a NAT rule that uses the PLAT IPv6 prefix for the destination address and destination prefix and uses the NAT translation type `stateful-nat464`. For the source address and CLAT prefix in the NAT rule, identify the IPv6 prefix for the CLAT. The NAT rule must specify a NAT pool that the PLAT uses for converting the private IPv4 source address to a public IPv4 address.

Benefits of 464XLAT

- No need to maintain an IPv4 transit network
- No need to assign additional public IPv4 addresses

Configuring 464XLAT Provider-Side Translater for IPv4 Connectivity Across IPv6-Only Network

Starting in Junos OS Release 17.1R1, you can configure a 464XLAT Provider-Side Tranlator (PLAT). This is supported only on MS-MICs and MS-MPCs. 464XLAT provides a simple and scalable technique for an IPv4 client with a private address to connect to an IPv4 host over an IPv6 network. 464XLAT only suports IPv4 in the client-server model, so it does not support IPv4 peer-to-peer communication or inbound IPv4 connections.

The following restrictions apply when configuring the PLAT:

- An `overload-pool` cannot be configured in the NAT rule.
- Different terms in the NAT rule cannot have the same `destination-prefix`.

To configure the PLAT:

1. Configure a NAT pool NAT pool that the PLAT uses for converting the private IPv4 source address to a public IPv4 address. See "Configuring Pools of Addresses and Ports for Network Address Translation Overview" on page 94 .

2. Configure a name for a NAT rule.

```
[edit services nat]
user@host# set rule rule-name
```

3. Configure a match direction for the rule. See "Network Address Translation Rules Overview" on page 97 .

4. Configure the IPv6 source address prefix. This must be the CLAT IPv6 prefix or contain the CLAT IPv6 prefix.

```
[edit services nat rule rule-name term term-name from]
user@host# set source-address address
```

5. Configure the IPv6 destination address prefix, which must have a length of /96. This is the PLAT destination IPv6 IP prefix.

```
[edit services nat rule rule-name term term-name from]
user@host# set destination-address address
```

6. Specify the NAT pool that the PLAT uses for converting the private IPv4 source address to a public IPv4 address.

```
[edit services nat rule rule-name term term-name then translated]
user@host# set source-pool nat-pool-name
```

7. Specify the CLAT IPv6 source prefix.

```
[edit services nat rule rule-name term term-name then translated]
user@host# set clat-prefix clat-prefix
```

8. Configure the IPv6 destination prefix, which must have a length of /96. This is the PLAT destination IPv6 IP prefix.

```
[edit services nat rule rule-name term term-name then translated]
user@host# set destination-prefix destination-prefix
```

9. Configure the translation type as stateful NAT464.

```
[edit services nat rule rule-name term term-name then translated]
user@host# set translation-type stateful-nat464
```

10. Enable address pooling paired (APP).

```
[edit services nat rule rule-name term term-name then translated]
user@host# set address-pooling paired.
```

11. Assign the NAT rule to a service set.

```
[edit services]
user@host# set service-set service-set-name nat-rules rule-name
```

CHAPTER 11

Port Control Protocol

IN THIS CHAPTER

- Port Control Protocol | 243

Port Control Protocol

IN THIS SECTION

- Port Control Protocol Overview | 243
- Configuring Port Control Protocol | 246
- Monitoring Port Control Protocol Operations | 252
- Example: Configuring Port Control Protocol with NAPT44 | 254

Port Control Protocol Overview

IN THIS SECTION

- Benefits of Port Control Protocol | 245
- Port Control Protocol Version 2 | 245

Port Control Protocol (PCP) provides a way to control the forwarding of incoming packets by upstream devices, such as NAT44 and firewall devices, and a way to reduce application keepalive traffic. PCP is supported on the MS-DPC, MS-100, MS-400, and MS-500 MultiServices PICs. Starting in Junos OS Release 17.4R1, PCP for NAPT44 is also supported on the MS-MPC and MS-MIC. Starting in Junos 20.2R1, PCP for CGNAT DS-Lite services are supported for Next Gen Services.Starting in Junos OS

Release 18.2R1, PCP on the MS-MPC and MS-MIC supports DS-Lite. In Junos OS Release 18.1 and earlier releases, PCP on the MS-MPC and MS-MIC does not support DS-Lite.

PCP is designed to be implemented in the context of both Carrier-Grade NATs (CGNs) and small NATs (for example, residential NATs). PCP enables hosts to operate servers for a long time (as in the case of a webcam) or a short time (for example, while playing a game or on a phone call) when behind a NAT device, including when behind a CGN operated by their ISP. PCP enables applications to create mappings from an external IP address and port to an internal IP address and port. These mappings are required for successful inbound communications destined to machines located behind a NAT or a firewall. After a mapping for incoming connections is created, remote computers must be informed about the IP address and port for the incoming connection. This is usually done in an application-specific manner.

Junos OS supports PCP version 2 and version 1.

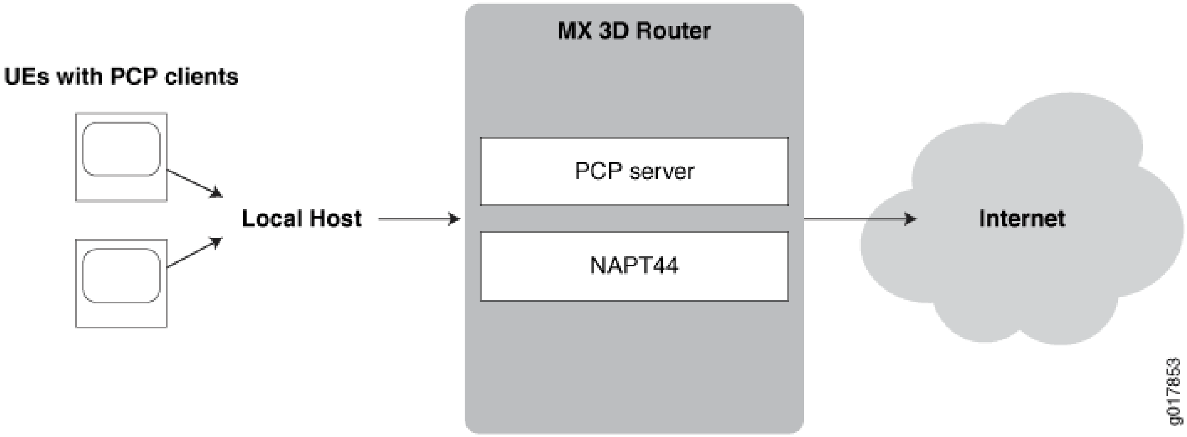
PCP consists of the following components:

- PCP client—A host or gateway that issues PCP requests to a PCP server in order to obtain and control resources.
- PCP server—Typically a CGN gateway or co-located server that receives and processes PCP requests

Junos OS enables configuring PCP servers for mapping flows using NAPT44 capabilities such as port forwarding and port block allocation. Flows can be processed from these sources:

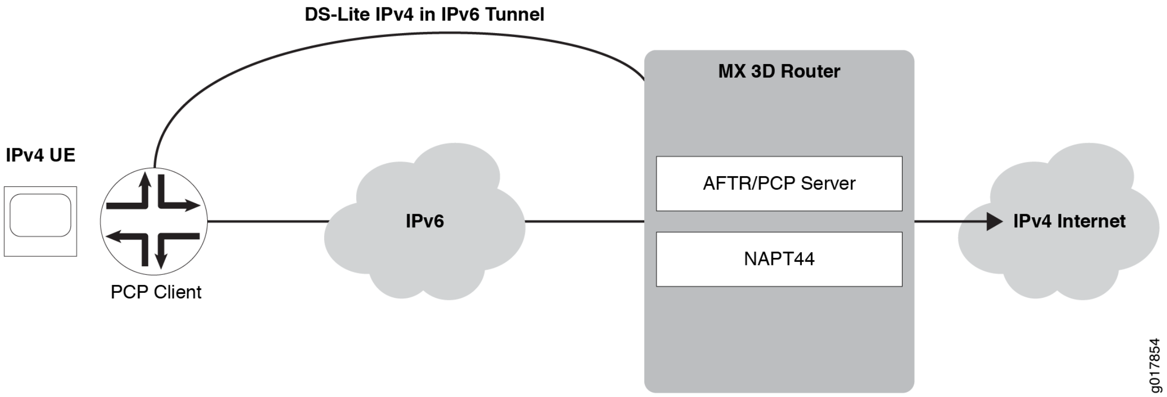
- Traffic containing PCP requests received directly from user equipment, as shown in Figure 13 on page 244 .

Figure 13: Basic PCP NAPT44 Topology



- Mapping of traffic containing PCP requests added by a router functioning as a DS-Lite softwire initiator (B4). This mode, known as *DS-Lite plain mode*, is shown in Figure 14 on page 245 .

Figure 14: PCP with DS-Lite Plain Mode



> **NOTE**: Junos OS does not support deterministic port block allocation for PCP-originated traffic.

Benefits of Port Control Protocol

Many NAT-friendly applications send frequent application-level messages to ensure their sessions are not being timed out by a NAT device. PCP is used to:

- Reduce the frequency of these NAT keepalive messages
- Reduce bandwidth on the subscriber's access network
- Reduce traffic to the server
- Reduce battery consumption on mobile devices

Port Control Protocol Version 2

Starting with Junos OS Release 15.1, Port Control Protocol (PCP) version 2 is supported, which is in compliance with RFC 6887. PCP provides a way to control the forwarding of incoming packets by upstream devices, such as NAT44, and firewall devices, and a way to reduce application keep-alive traffic. PCP version 2 supports nonce authentication. PCP allows applications to create mappings from an external IP address and port to an internal IP address and port. A nonce payload prevents a replay attack and it is sent by default unless it is explicitly disabled.

Client nonce verification for version 2 map requests (for refresh or delete) requires that the nonce received in the original map request that causes the PCP mapping to be created is preserved. The version of the initial request that enables the mapping to be created is also preserved. This behavior of saving the nonce and version parameters denotes that 13 bytes per PCP mapping are used. This slight increase in storage space is not significant when matched with the current memory usage of a system for a single requested mapping (taking into account the endpoint-independent mapping (EIM) and endpoint-independent filtering (EIF) that are created along with it). In a customer deployment, PCP causes EIM and EIF mappings to represent a fraction of all such mappings.

Until Junos Release 15.1, services PICs support PCP servers on Juniper Networks routers in accordance with PCP draft version 22 with version 1 message encoding. With PCP being refined from the draft version as defined in *Port Control Protocol (PCP) draft-ietf-pcp-base-22 (July 2012 expiration)* to a finalized, standard version as defined in RFC 6887 -- Port Control Protocol (PCP), the message encoding changed to version 2 with the addition of a random nonce payload to authenticate peer and map requests as necessary. Version 1 does not decode messages compliant with version 2 format and nonce authentication is not supported. In a real-word network environment, with customer premises equipment (CPE) devices increasingly supporting version 2 only, it is required to parse and send version 2 messages. Backward compatibility with version 1-supporting CPE devices is maintained (version negotiation is part of the standard) and authenticates request nonce payload packets when v2 messages are in use.

The output of the `show services pcp statistics` command contains the PCP unsupported version field, which is incremented to indicate whenever the version is not 1 or 2. A new field, PCP request nonce does not match existing mapping, is introduced to indicate the number of PCP version 2 requests that were ignored because the nonce payload did not match the one recorded in the mapping (authentication failed). If version 2 is in use, the client nonce is used for authentication.

Configuring Port Control Protocol

IN THIS SECTION

- Configuring PCP Server Options | **247**
- Configuring a PCP Rule | **249**
- Configuring a NAT Rule | **250**
- Configuring a Service Set to Apply PCP | **251**
- SYSLOG Message Configuration | **252**

This topic describes how to configure port control protocol (PCP). PCP is supported on the MS-DPC, MS-100, MS-400, and MS-500 MultiServices PICs. Starting in Junos OS Release 17.4R1, PCP for NAPT44 is also supported on the MS-MPC and MS-MIC. Starting in Junos OS Release 18.2R1, PCP on the MS-MPC and MS-MIC supports DS-Lite. In Junos OS Release 18.1 and earlier releases, PCP on the

MS-MPC and MS-MIC does not support DS-Lite. Starting in Junos OS release 20.2R1 PCP is supported on the MX-SPC3 security services card for CGNAT services.

Perform the following configuration tasks:

Configuring PCP Server Options

1. Specify a PCP server name.

   ```
   user @host# edit services pcp server server-name
   ```

2. Set the IPv4 or IPv6 addresses of the server. For PCP DS-Lite, the `ipv6-address` must match the address of the AFTR (Address Family Transition Router or softwire concentrator).

   NOTE: Starting in Junos OS Release 18.2R1, PCP on the MS-MPC and MS-MIC supports DS-Lite. In Junos OS Release 18.1 and earlier releases, PCP on the MS-MPC and MS-MIC does not support DS-Lite.

   ```
   [edit services pcp server server-name]
   user @host# set ipv6-address ipv6-address
   ```

   or

   ```
   [edit services pcp server server-name]
   user @host# set ipv4-address ipv4-address
   ```

3. For PCP DS-Lite, provide the name of the DS-Lite softwire concentrator configuration.

   ```
   [edit services pcp server server-name]
   user @host# set softwire-concentrator softwire-concentrator-name
   ```

4. Specify the minimum and maximum mapping lifetimes for the server.

   ```
   [edit services pcp server server-name]
   user @host# set mapping-lifetime-minimum mapping-lifetime-min
   user @host# set mapping-lifetime-maximum mapping-lifetime-max
   ```

5. Specify the time limits for generating short lifetime or long lifetime errors.

```
[edit services pcp server server-name]
user @host# set short-lifetime-error short-lifetime-error
user @host# set long-lifetime-error long-lifetime-error
```

6. (Optional)—Enable PCP options on the specified PCP server. The following options are available—`third-party` and `prefer-failure`. The third-party option is required to enable third-party requests by the PCP client. DS-Lite requires the `third-party` option. The `prefer-failure` option requests generation of an error message when the PCP client requests a specific IP address/port that is not available, rather than assigning another available address from the NAT pool. If `prefer-failure` is not specified NAPT44 assigns an available address/port from the NAT pool based on the configured NAT options.

```
[edit services pcp server server-name]
user @host# set pcp-options third-party
user @host# set pcp-options prefer-failure
```

7. (Optional)—Specify which NAT pool to use for mapping.

```
[edit services pcp server server-name]
user @host# set nat-options pool-name1 <poolname2...>
```

NOTE: When you do not explicitly specify a NAT pool for mapping, the Junos OS performs a partial rule match based on source IP, source port, and protocol, and the Junos OS uses the NAT pool configured for the first matching rule to allocate mappings for PCP.
You *must* use explicit configuration in order to use multiple NAT pools.

For the MX-SPC3 security services card and Next Gen Services, the `nat-options` statement supports only one pool name to attach to a PCP server.

8. (Optional)—Configure the maximum number of mappings per client. The default is 32 and maximum is 128.

```
[edit services pcp server server-name]
user @host# set max-mappings-per-client max-mappings-per-client
```

Configuring a PCP Rule

A PCP rule has the same basic options as all service set rules:

- A `term` option that allows a single rule to have multiple applications.

  A term is not required when running the MX-SPC3 security services card for Next Gen Services.

- A `from` option that identifies the traffic that is subject to the rule.

- A `then` option that identifies what action is to be taken. In the case of a PCP rule, this option Identifies the pcp server that handles selected traffic

1. Go to the `[edit services pcp rule` *`rule-name`*`]` hierarchy level and specify `match-direction` input.

```
user @host# edit services pcp rule rule-name
user @host# set match-direction input
```

2. Go to the `[edit services pcp rule` *`rule-name`* `term` *`term-name`*`]` hierarchy level and provide a term name.

```
user @host# edit term term-name
```

   This step is not required when running the MX-SPC3 security services card for Next Gen Services.

3. (Optional)—Provide a `from` option to filter the traffic to be selected for processing by the rule. When you omit the `from` option, all traffic handled by the service set's service interface is subject to the rule. The following options are available at the `[edit services pcp rule` *`rule-name`* `term` *`term-name`* `from]` hierarchy level:

   `application-sets` ***`set-name`***

   Traffic for the application set is processed by the PCP rule.

   This step is not required when running the MX-SPC3 security services card for Next Gen Services.

   `applications` **`[`** ***`application-name`*** **`]`**

   Traffic for the application is processed by the PCP rule.

   This option is not required when running the MX-SPC3 security services card for Next Gen Services.

   `destination-address` ***`address`*** **`<except>`**

   Traffic for the destination address or prefix is processed by the PCP rule. If you include the `except` option, traffic for the destination address or prefix is *not* processed by the PCP rule.

destination-address-range **high** ***maximum-value*** **low** ***minimum-value*** **<except>**	Traffic for the destination address range is processed by the PCP rule. If you include the except option, traffic for the destination address range is *not* processed by the PCP rule.
destination-port **high** ***maximum-value*** **low** ***minimum-value***	Traffic for the destination port range is processed by the PCP rule.
destination-prefix-list ***list-name*** **<except>**	Traffic for a destination address in the prefix list is processed by the PCP rule. If you include the except option, traffic for a destination address in the prefix list is *not* processed by the PCP rule.
source-address ***address*** **<except>**	Traffic from the source address or prefix is processed by the PCP rule. If you include the except option, traffic from the source address or prefix is *not* processed by the PCP rule.
source-address-range **high** ***maximum-value*** **low** ***minimum-value*** **<except>**	Traffic from the source address range is processed by the PCP rule. If you include the except option, traffic from the source address range is *not* processed by the PCP rule.
source-prefix-list ***list-name*** **<except>**	Traffic from a source address in the prefix list is processed by the PCP rule. If you include the except option, traffic from a source address in the prefix list is *not* processed by the PCP rule.

4. Set the then option to identify the target PCP server.

```
[edit services pcp rule rule-name term term-name]
user @host# set then pcp-server server-name
```

Configuring a NAT Rule

To configure a NAT rule:

1. Configure the NAT rule name and the match direction.

```
 [edit services nat]
user@host# set rule rule-name match-direction match-direction
```

2. Specify the NAT pool to use:

```
 [edit services nat rule-name term term-name then translated]
user@host# set source-pool nat-pool-name
```

3. Configure the translation type.

```
 [edit services nat rule-name term term-name then  translated]
user@host# set translation-type translation-type
```

4. If you are using PCP with IPv4-to-IPv4 NAT or with DS-Lite, configure endpoint-independent mapping (EIM) and endpoint-independent filtering (EIF).

```
 [edit services nat rule-name term term-name then  translated]
user@host# set mapping-type endpoint-independent
user@host# set filtering-type endpoint-independent
```

NOTE: The PCP mappings are not created if you do not configure EIM and EIF with PCP for IPv4-to-IPv4 NAT or for DS-Lite.

Configuring a Service Set to Apply PCP

To use PCP, you must provide the rule name (or name of a list of rule names) in the `pcp-rule` *`rule-name`* option.

1. Go to the `[edit services service-set` *`service-set-name`* hierarchy level.

```
user @host# edit services service-set service-set-name
```

2. If this is a new service set, provide basic service set information, including interface information and any other rules that may apply.
3. Specify the name of the PCP rule or rule list used to send traffic to the specified PCP server.

```
[edit services service-set service-set-name ]
user @host# set pcp-rule rule-name | rule-listname
```

NOTE: Your service set must also identify any required `nat-rule` and `softwire-rule`.

SYSLOG Message Configuration

A new syslog class, configuration option, `pcp-logs`, has been provided to control PCP log generation. It provides the following levels of logging:

- `protocol`—All logs related to mapping creation, deletion are included at this level of logging.
- `protocol-error`—–All protocol error related logs (such as mapping refresh failed, PCP look up failed, mapping creation failed). are included in this level of logging.
- `system-error`—Memory and infrastructure errors are included in this level of logging.

Monitoring Port Control Protocol Operations

You can monitor Port Control Protocol (PCP) operations with the following operational commands:

- For MS-MPCs use the `show services nat mappings pcp` command.

  NOTE: PCP is not supported for Next Gen Services in Junos OS Release 19.3R2

- For MS-MPCs use the `show services nat mappings endpoint-independent` command.

  For Next Gen Services use the `show services nat source mappings endpoint-independent` command.

- `show services pcp statistics protocol`

The following are examples of the output of these commands.

```
user@host> show services nat mappings pcp
Interface: sp-0/0/0, Service set: in

NAT pool: p
PCP Client       : 10.1.1.2
PCP lifetime     : 995
Mapping          : 10.1.1.2        : 9000  --> 8.8.8.8         : 1025
Session Count    :     1
Mapping State    : Active

DS-LITE output:
===============
PCP Client       : 2222::1
PCP lifetime : 106
Mapping          : 88.1.0.47       :   47  --> 70.70.70.1      :41972
Session Count    :     1
```

```
Mapping State    : Active
B4 Address       : 2222::1
```

```
user@host> show services nat mappings endpoint-independent
Interface: sp-0/0/0, Service set: in

NAT pool: p
Mapping          : 10.1.1.2        :57400  --> 8.8.8.8         : 1024
Session Count    :     0
Mapping State    : Timeout
PCP Client       : 10.1.1.2                 PCP lifetime : 991
Mapping          : 10.1.1.2        : 9000  --> 8.8.8.8         : 1025
Session Count    :     1
Mapping State    : Active

DS-LITE output:
===============
PCP Client       : 2222::1                 PCP lifetime : 190
Mapping          : 88.1.1.3        : 4001  --> 70.70.70.2      :58989
Session Count    :     1
Mapping State    : Active
B4 Address       : 2222::1
```

```
user@host> show services pcp statistics protocol
Protocol Statistics:

Operational Statistics

  Map request received               :0
  Peer request received              :0
  Other operational counters         :0

Option Statistics

  Unprocessed requests received      :0
  Third party requets received       :0
  Prefer fail option received        :0
  Filter option received             :0
  Other options counters             :0
  Option optional received           :0
```

```
Result Statistics

  PCP success                        :0
  PCP unsupported version            :0
  Not authorized                     :0
  Bad requests                       :0
  Unsupported opcode                 :0
  Unsupported option                 :0
  Bad option                         :0
  Network failure                    :0
  Out of resources                   :0
  Unsupported protocol               :0
  User exceeded quota                :0
  Cannot provide external            :0
  Address mismatch                   :0
  Excessive number of remote peers   :0
  Processing error                   :0
  Other result counters              :0
```

Example: Configuring Port Control Protocol with NAPT44

IN THIS SECTION

- Requirements | 254
- Overview | 255
- PCP Configuration | 255

NOTE: PCP is supported on the MS-DPC, MS-100, MS-400, and MS-500 MultiServices PICS. Starting in Junos OS Release 17.4R1, PCP for NATP44 is also supported on the MS-MPC and MS-MIC.

Requirements

Hardware Requirements

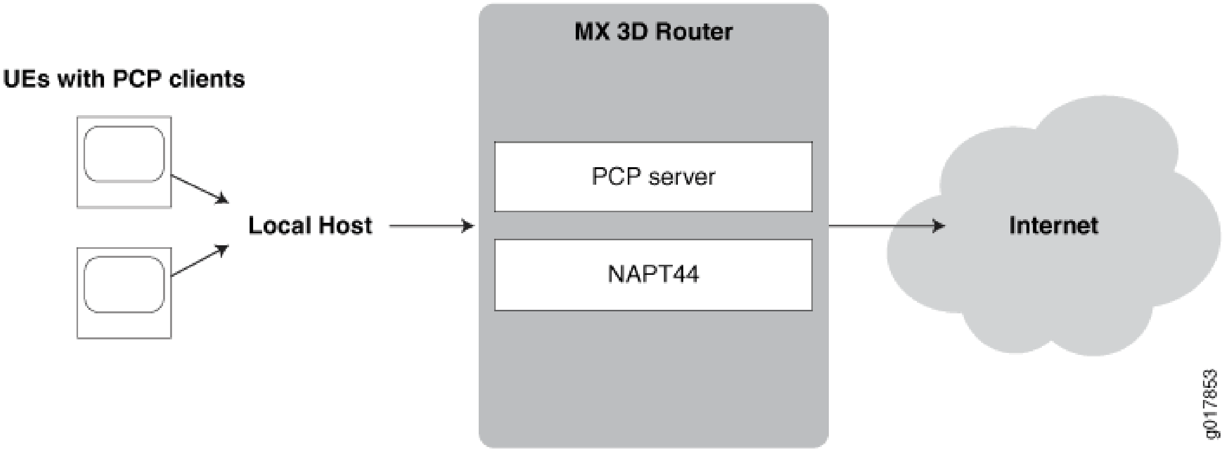
- UEs with PCP clients.

- An MX 3D Router with an MS-DPC services PIC.
- Software Requirements
- Junos OS 13.2
- Layer-3 Services Package

Overview

An ISP wants to enable UEs with PCP clients to maintain connections to servers without timing out. The PCP clients generate PCP requests for the type and duration of the connection they require. Connections may be of a long duration, such as applications using a webcam, or a shorter duration, such as online games. An MX 3D router provides a PCP server to interpret PCP client requests, and NAPT44. Figure 15 on page 255 shows the basic topology for this example.

Figure 15: PCP with NAPT44



PCP Configuration

IN THIS SECTION

- CLI Quick Configuration | 256
- Chassis Configuration | 256
- Interface Configuration | 257
- NAT Configuration | 259
- PCP Configuration | 260

- Service Set Configuration | 261

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the [edit] hierarchy level.

```
set chassis fpc 2 pic 0 adaptive-services service-package layer-3
set interfaces sp-2/0/0 services-options inactivity-timeout 180 cgn-pic
set interfaces sp-2/0/0 unit 0 family inet
set interfaces xe-3/2/0 unit 0 family inet service input service-set sset_0
set interfaces  xe-3/2/0 unit 0 family inet service output service-set sset_0
set interfaces  xe-3/2/0 unit 0 family inet address 30.0.0.1/24
set interfaces xe-5/0/0 unit 0 family inet address 25.0.0.1/24
set services nat pool pcp-pool address 44.0.0.0/16
set services nat pool pcp-pool port automatic random-allocation address-allocation round-robin
set services nat pool pcp-pool address-allocation round-robin
set services nat rule pcp-rule match-direction input
set services nat rule pcp-rule term t0 then translated source-pool pcp-pool translation-type
napt-44
set services nat rule pcp-rule term t0 then translated mapping-type endpoint-independent
filtering-type endpoint-independent
set services nat rule pcp-rule term t0 then translated mapping-type endpoint-independent
filtering-type endpoint-independent
set services pcp server pcp-s1 ipv4-address 124.124.124.122
set services pcp server pcp-s1 mapping-lifetime-minimum 600 mapping-lifetime-maximum 86500
set services pcp server pcp-s1 short-lifetime-error 120 long-lifetime-error 1200
set services pcp server pcp-s1 max-mappings-per-client 128 pcp-options third-party prefer-failure
set services service-set sset_0 pcp-rules r1
set services service-set sset_0 nat-rules pcp-rule
set services service-set sset_0 interface-service service-interface sp-2/0/0.0
```

Chassis Configuration

Step-by-Step Procedure

To configure the service PIC (FPC 2 Slot 0) with the Layer 3 service package:

1. Go to the [edit chassis] hierarchy level.

```
user@host# edit chassis
```

2. Configure the Layer 3 service package.

```
[edit chassis]
user@host# set fpc 2 pic 0 adaptive-services service-package layer-3
```

Results

```
user@host# show chassis fpc 2 pic 0

pcp-rules pcp-napt44-rule;
nat-rules pcp-rule;
interface-service {
    service-interface sp-2/0/0.0;
}
```

Interface Configuration

Step-by-Step Procedure

1. Configure the services MS-DPC.

```
user@host# set interfaces sp-2/0/0 services-options inactivity-timeout 180 cgn-pic
user@host# set interfaces sp-2/0/0 unit 0 family inet
```

2. Configure the customer-facing interface used for NAT and PCP services.

```
user@host# set interfaces xe-3/2/0 unit 0 family inet service input service-set sset_0
user@host# set interfaces  xe-3/2/0 unit 0 family inet service output service-set sset_0
user@host# set interfaces  xe-3/2/0 unit 0 family inet address 30.0.0.1/24
```

3. Configure the Internet-facing interface.

```
user@host# set interfaces xe-5/0/0 unit 0 family inet address 25.0.0.1/24
```

Results

```
user@host#
sp-2/0/0 {
    services-options {
        inactivity-timeout 180;
        cgn-pic;
    }
    unit 0 {
        family inet;
    }
}
xe-3/2/0 {
    unit 0 {
        family inet {
            service {
                input {
                    service-set sset_0;
                }
                output {
                    service-set sset_0;
                }
            }
            address 30.0.0.1/24;
        }
    }
}
xe-5/0/0 {
    unit 0 {
        family inet {
            address 25.0.0.1/24;
        }
    }
}
```

NAT Configuration

Step-by-Step Procedure

1. Go the `[edit services nat]` hierarchy.

```
user@host# edit services nat
```

2. Configure a NAT pool called `pcp-pool`.

```
[edit services nat]
user@host# set pool pcp-pool address 44.0.0.0/16
user@host# set pool pcp-pool port automatic random-allocation
user@host# set pool pcp-pool address-allocation round-robin
```

3. Configure a NAT rule called `pcp-rule`.

```
[edit services nat]
user@host# set rule pcp-rule term t0 then translated source-pool pcp-pool translation-type
napt-44
user@host# set rule pcp-rule term t0 then translated mapping-type endpoint-independent
filtering-type endpoint-independent
```

Results

```
user@host# show services nat
pool pcp-pool {
    address 44.0.0.0/16;
    port {
        automatic {
            random-allocation;
        }
    }
    address-allocation round-robin;
}
rule pcp-rule {
    match-direction input;
    term t0 {
        then {
            translated {
                source-pool pcp-pool;
                translation-type {
                    napt-44;
                }
                mapping-type endpoint-independent;
                filtering-type {
                    endpoint-independent;
                }
            }
        }
    }
}
```

PCP Configuration

Step-by-Step Procedure

To configure the PCP server and PCP rule options.

1. Go to the `edit services pcp` hierarchy level for server `pcp-s1`

    ```
    user@host# edit services pcp server pcp-s1
    ```

2. Configure the PCP server options.

    ```
    [edit services pcp server pcp-s1]
    user@host# set ipv4-address 124.124.124.122
    user@host# set mapping-lifetime-minimum 600
    user@host# set mapping-lifetime-maximum 86500
    user@host# set short-lifetime-error 120
    user@host# set long-lifetime-error 1200
    user@host# set max-mappings-per-client 128
    user@host# set pcp-options third-party prefer-failure
    ```

3. Create the PCP rule.

```
[edit services pcp rule pcp-napt44-rule
user@host# edit rule pcp-napt44-rule
```

4. Configure the PCP rule options.

```
[edit services pcp rule pcp-napt44-rule]
user@host# set match-direction input
user@host# set term t0 then pcp-server pcp-s1
```

Results

```
user@host# show services pcp

server pcp-s1 {
    ipv4-address 124.124.124.122;
    mapping-lifetime-minimum 600;
    mapping-lifetime-maximum 86500;
    short-lifetime-error 120;
    long-lifetime-error 1200;
    max-mappings-per-client 128;
    pcp-options third-party prefer-failure;
}
rule pcp-napt44-rule {
    match-direction input;
    term t0 {
        then {
            pcp-server pcp-s1;
        }
    }
}
```

Service Set Configuration

Step-by-Step Procedure

1. Create a service set, `sset_0`, at the `edit services service-set` hierarchy level.

   ```
   user@host# edit services service-set sset_0
   ```

   ```
   service-set sset_0 {
       pcp-rules pcp-napt44-rule;
       nat-rules pcp-rule;
       interface-service {
           service-interface sp-2/0/0.0;
       }
   }
   ```

2. Identify the NAT rule associated with the service set.

   ```
   [edit services service-set sset_0]
   user@host# set nat-rules pcp-rule
   ```

3. Identify the PCP rule associated with the service set.

   ```
   [edit services service-set sset_0]
   user@host# set pcp-rules r1
   ```

4. Identify the service interface associated with the service set.

   ```
   [edit services service-set sset_0]
   user@host# set interface-service service-interface sp-2/0/0.0
   ```

Results

```
user@host# show
pcp-rules pcp-napt44-rule;
nat-rules pcp-rule;
interface-service {
    service-interface sp-2/0/0.0;
}
```

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
20.2R1	Starting in Junos 20.2R1, PCP for CGNAT DS-Lite services are supported for Next Gen Services.
20.2R1	Starting in Junos OS release 20.2R1 PCP is supported on the MX-SPC3 security services card for CGNAT services.
18.2R1	Starting in Junos OS Release 18.2R1, PCP on the MS-MPC and MS-MIC supports DS-Lite.
18.2R1	
17.4R1	Starting in Junos OS Release 17.4R1, PCP for NAPT44 is also supported on the MS-MPC and MS-MIC.
17.4R1	Starting in Junos OS Release 17.4R1, PCP for NAPT44 is also supported on the MS-MPC and MS-MIC.
17.4R1	Starting in Junos OS Release 17.4R1, PCP for NATP44 is also supported on the MS-MPC and MS-MIC.
15.1	Starting with Junos OS Release 15.1, Port Control Protocol (PCP) version 2 is supported, which is in compliance with RFC 6887.

CHAPTER 12

Secured Port Block Allocation

IN THIS CHAPTER

- Secured Port Block Allocation | 264
- Secured Port Block Allocation Interim Logging | 271

Secured Port Block Allocation

IN THIS SECTION

- Secured Port Block Allocation for NAPT44 and NAT64 Overview | 264
- Guidelines for Configuring Secured Port Block Allocation | 265
- Configuring Secured Port Block Allocation | 267

Secured Port Block Allocation for NAPT44 and NAT64 Overview

IN THIS SECTION

- Benefits of Secured Port Block Allocation | 265

Secured port block allocation ensures that when a subscriber requires a port to be assigned for the first time, a block of ports are allocated to the particular user. Here, a subscriber is defined uniquely as a private IP address and service set ID. Because the subscriber has a block of ports assigned to it, all subsequent requests from this subscriber use ports from the assigned block. A new port block is allocated when the current active block is exhausted, or after the active port block timeout interval has expired. You can configure the maximum number of blocks allocated to a user. This behavior of allocation of NAT ports in blocks is different from the traditional NAT utility where the request for a port allocates a single port and not a group of ports in a block.

You can use the secured port block allocation mechanism to allocate ports in blocks for NAPT44 (translation of an IPv4 address to an IPv4 address) and NAT64 (translation of an IPv6 address to an IPv4 address) types. By using secured port block allocation, the port usage might be a little inefficient, depending on traffic patterns. Secured port block allocation is supported on MX series routers with MS-DPCs and on M Series routers with MS-100, MS-400, and MS-500 MultiServices PICS. Starting in Junos OS release 14.2R2, secured port block allocation is supported on MX series routers with MS-MPCs and MS-MICs.

Starting with Junos OS Release 15.1, in an environment in which Junos Address Aware (carrier-grade NAT) is employed, service providers or carrier operators can monitor and track the consumption of resources and types of services being utilized by subscribers or users in an easier and effective manner by using system logging messages recorded for the allocation of ports to clients. By using IP addresses in RADIUS or DHCP logs, evaluation of the logs is performed to analyze an determine the services usage and bandwidth consumption by subscribers. With carrier-grade NAT, because IP addresses are shared by multiple subscribers, examining logs to track the IP addresses and ports that are part of the system logs might be time-consuming and difficult. Also, because ports are allocated and released at frequent intervals depending on the logging-in and closure of subscriber sessions, a large number of logs are triggered for every port allocation and deallocation. As a result, excessive syslogs render it cumbersome to archive and correlate the logs to identify a subscriber. You can now allocate ports in blocks, which reduces the amount of syslogs considerably.

Benefits of Secured Port Block Allocation

- Reduces the effort to correlate logs to a subscriber
- Reduces the number of logs

Guidelines for Configuring Secured Port Block Allocation

Keep the following points in mind when you configure secured PBA:

- Block size is not configurable at the NAT rule level.
- Increase in setup rate of sessions is not impacted when you configure secured PBA.
- If a block of a particular size is not available, an out-of-ports message is displayed and smaller-sized blocks are not allocated alternatively in such a scenario.
- Addresses in the pool using port-block-allocation method cannot be used in any other pool.
- Port range in the NAT pool must be contiguous.

- Preserve parity (Allocate ports with same parity as the original port) is not supported with block-allocation of ports.

- The limitation on the number of open sessions when the specified threshold is reached (for intrusion detection services) and the maximum number of blocks that can be allocated to a user address that is configured for secured PBA are independent functionalities.

- The functionality to preserve privileged port range after translation is not supported. The blocks are assigned from unprivileged port range (1024-65535). For ports in privileged range, port block allocation method is not applicable.

- Port usage efficiency is lower when port-block allocation is enabled. PBA does not use ports from 0-1023 of a NAT IP address.

- If you configure the automatic port assignment method, which enables sequential assignment of ports, the port range from 1024 through 65535 is available for allocation to users.

- Port blocks can start at any start port that you can configure.

- The number of ports used is dependent on the block size and the rest of the ports are not be used.

- An overloaded pool, which indicates an address pool that can be used if the source pool becomes exhausted, is not supported with secured PBA.

- NAT IP addresses of PBA pool must not overlap with any other pool. Although a validation is not performed to identify whether any overlapping pools exist, you must ensure that the addresses of a pool that is used for PBA are not used in other pools. This condition is because some of the users require the overload pool to use the same IP addresses as that of NAT IP addresses, but a different port range of PBA pool to support the address pooling paired (APP) functionality.

- The block-size is fixed per NAT pool and is configurable at the NAT pool level. Multiple port blocks can be allocated to a private IP address.

- You can configure the maximum number of blocks per pool per subscriber by including the `max-blocks-per-user` *`max-blocks`* statement at the `[edit services nat pool pool-name port secured-port-block-allocation]` hierarchy level. If a subscriber matches two pools, that particular user can be allocated a maximum of port blocks that equals the sum of the maximum number of port blocks for each pool for that subscriber. New requests for NAT ports arrive from the current active block only.

- Ports can be allocated randomly from the current active block, which specifies whether ports should be allocated sequentially or randomly within the port block.

- A block is active for a timeout interval that you can define by including the `active-block-timeout` *`timeout-seconds`* at the `[edit services nat pool pool-name port secured-port-block-allocation]` hierarchy level. After the timeout period, a new block is allocated even if ports are available in the active block. The default timeout of an active block is 120 seconds. When you configure it as 0 (infinite), the active block transitions to inactive only when it runs out of ports and a new block is allocated.

- If the maximum number of blocks of blocks is exceeded, and a new request is received, the active block is moved to a block that contains available ports. Any non-active block without any ports in use is freed to NAT pool.

- In addition to tracking port blocks assigned to each private IP address, actual ports in use are also computed and maintained. This metric is used to calculate port usage efficiency.

- A syslog message is generated for each block allocation and release. The format of the message is similar to the messages recorded for individual port allocation and release.

- Session setup rate is the same or slightly improved than the existing non-block allocation setup rate. NAT pool using block-port allocation method can have partial port ranges. If the address is used for port forwarding, those ports can be removed from the pool port range. You can configure partial port ranges by using the `port range low minimum-value high maximum-value random-allocation` statement at the `[edit services nat pool nat-pool-name]` hierarchy level. Port block allocation works in the same manner as NAPT44 for TCP, UDP, and ICMP traffic.

- Randomness can be achieved by allocating ports randomly within the block and changing active block periodically. The block of ports do not contain random ports (ports within the block are sequential). This capability is supported with aggregated multiservices (ams) interfaces.

- The starting port number is calculated differently in the microkernel and in Junos OS Extension-Provider packages. In the microkernel, the starting or first port is the nearest multiple of the block size after 1023. In that implementation, more ports are wasted because ports are wasted at the beginning and the end of the port range depending on the block size. In Junos OS Extension-Provider packages, the start port of a block is not restricted to a multiple of the block size. The start port can start at the lower boundary of the range of the port configured.

SEE ALSO

Configuring NAT Session Logs | 347

Configuring Secured Port Block Allocation

Secured port block allocation is supported on MX series routers with MS-DPCs and on M Series routers with MS-100, MS-400, and MS-500 MultiServices PICS. Secured port block allocation is supported on MX series routers with MS-MPCs and MS-MICs starting in Junos OS release 14.2R2. To configure secured port block allocation:

1. At the `[edit services nat pool nat-pool-name]` hierarchy level, create a pool.

```
user@host# edit services nat pool poolname
```

For example:

```
user@host# edit services nat pool pba-pool1
```

2. Define the range of addresses to be translated, specifying the upper and lower limits of the range or an address prefix that describes the range.

```
[edit services nat pool nat-pool-name]
user@host# set address-range low address high address
```

Or

```
user@host# set address address-prefix
```

For example:

```
[edit services nat pool pba-pool1]
user@host# set address 203.0.113.0/24
```

3. Define the range of ports to be used in the translation, or use automatic port assignment by the Junos OS. You can optionally specify random assignment of ports; sequential assignment is the default.

```
[edit services nat pool nat-pool-name]
user@host# set port range low address high address random
```

Or

```
user@host# set port automatic random-allocation
```

For example:

```
[edit services nat pool pba-pool1]
user@host# set port range low 256 high 511 random
```

Or

```
[edit services nat pool pba-pool1]
user@host# set port automatic random-allocation
```

NOTE: When you configure a port range, the range should be a multiple of the port block-size value (see Step 4). When the `nat pool port range` is *not* a multiple of the port `block-size` value, the number of ports or port-blocks that are effectively available for use is less than the configured number of ports and port-blocks.

When you configure automatic assignment of ports, the available port range for allocation is 1024 through 65535. Automatic allocation can result in no ports being available for use. Use the `show services nat pool` command on the Routing Engine after you configure the port block allocation method to determine the number of ports and port blocks available for allocation to users.

4. Configure secured port block allocation. Specify `active-block-timeout`, `block-size`, and `max-blocks-per-address`, or accept the default values for those options.

```
[edit services nat pool nat-pool-name]
user@host# set secured-port-block-allocation active-block-timeout active-block-timeout block-size block-size max-blocks-per-address max-blocks-per-address
```

For example:

```
[edit services nat pool pba-pool1]
user@host# set secured-port-block-allocation active-block-timeout 120 block-size 256 max-blocks-per-address 12
```

NOTE: In order for `secured-port-block-allocation` configuration changes to take effect, you must reboot the services PIC whenever you change any of the following `nat pool` options:

- *`nat-pool-name`*
- `address` or `address-range`
- `port range`

- `port secured-port-block-allocation block-size`
- `port secured-port-block-allocation max-blocks-per-address.`
- `port secured-port-block-allocation active-block-timeout.`
- `from` hierarchy in the `nat rule`

NOTE: If you make any configuration changes related to a NAT pool that has secured port block allocation configured, you must delete the existing NAT address pool, wait at least 5 seconds, and then configure a new NAT address pool. We also strongly recommend that you perform this procedure if you make any changes to the NAT pool configuration, even when secured port block allocation is not configured.

NOTE: MS-MICs and MS-MPCs support up to a maximum of nine million port blocks per NPU. If your configuration exceeds this maximum supported number, one or more service sets might not be activated on that NPU.

SEE ALSO

Network Address Translation Configuration Overview | 92

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
14.2R2	Starting in Junos OS release 14.2R2, secured port block allocation is supported on MX series routers with MS-MPCs and MS-MICs.

Secured Port Block Allocation Interim Logging

IN THIS SECTION

- Interim Logging for Secured Port Block Allocation | 271
- Guidelines for Configuring Interim Logging for Secured Port Block Allocation | 272

Interim Logging for Secured Port Block Allocation

IN THIS SECTION

- Benefits of Iterim Logging | 271

With port block allocation we generate one syslog log per set of ports allocated for a subscriber. These logs are UDP based and can be lost in the network, particularly for long-running flows. Interim logging triggers re-sending the above logs at a configured interval for active blocks that have traffic on at least one of the ports of the block. Depending on your network topology, you can set the interval for the port block allocation logs based on the period of the archive so that at least one log per port block (for an active flow) in each archive is present.

To configure the interim logging interval at the services interface level, which applies to all the NAT pools on that ms- interface, include the `pba-interim-logging-interval` *seconds* statement at the `[edit interfaces ms-fpc/pic/port services-options]` hierarchy level. The `pba-interim-logging-interval` option is supported on MX series routers with MS-DPCs and on M Series routers with MS-100, MS-400, and MS-500 MultiServices PICS. The `pba-interim-logging-interval` option is supported on MX series routers with MS-MPCs and MS-MICs starting in Junos OS release 14.2R2.

Starting in Junos OS Release 15.1R1, you can also configure the interim logging interval at a NAT pool level. This capability is supported only on MX Series routers with MS-MPCs and MS-MICs. To configure the interim logging interval at a NAT pool level, include the `interim-logging-interval` *seconds* statement at the `[edit services nat pool` *pool-name* `port secured-port-block-allocation]` hierarchy level. You can specify a value from 0 through 86400 seconds for the interim logging frequency.

Benefits of Iterim Logging

- Enables you to identify the currently used port blocks

- Eliminates the need to search and analyze archived logs to identify the internal host that is using the external IP address and port

SEE ALSO

> Configuring NAT Session Logs | 347

Guidelines for Configuring Interim Logging for Secured Port Block Allocation

Observe the following guidelines when you configure the interim logging interval for secured port block allocation:

- Interim logging is enabled only when the interim logging functionality is configured. The `pba-interim-logging-interval` statement that you can configure at the `[edit interfaces` *`ms-fpc/pic/port`* `services-options]` hierarchy level of an ms-interface is provided for backward compatibility. The `pba-interim-logging-interval` option is supported on MX series routers with MS-DPCs and on M Series routers with MS-100, MS-400, and MS-500 MultiServices PICS. The `pba-interim-logging-interval` option is supported on MX series routers with MS-MPCs and MS-MICs starting in Junos OS release 14.2R2.

  The `interim-logging-interval` statement that is available for configuration on the MS-MPC and MS-MIC starting in Junos OS release 15.1R1 provides interim logging for a specific NAT pool.

- If you configure the interim logging capability to be applicable to all PBA pools residing on that particular services interface and the interim logging capability for a specific PBA pool, the NAT pool-specific interval takes precedence over the services interface specific interval. For port blocks allocated from other PBA pools for which interim logging interval at the NAT pool-level is not configured, the logging interval value as configured at the ms- interface-level applies.

- The default value is zero, which denotes no interim logging message is generated.

- Interim logs are sent any time after the configured period of time in seconds. The time-difference is not fixed between the logging intervals of two logs.

- Interim logs are generated for port blocks (both active and inactive) that contain at least one port in use by a flow which has traffic. No timer controls run on the port blocks to generate the logs. When a packet is received on a flow, the validation is performed to generate an interim log. If the conditions are satisfied, an interim log is generated for that port block. Interim logs are not generated for deleted port blocks.

- The interim log contains the timestamp of the port block creation in hexadecimal format (when local time is set, the hexadecimal value provides the time in UTC format).

- The conversion of the timestamp to UTC format can be performed in the external syslog server as necessary.

- In certain scenarios, it is possible that the timestamp in hexadecimal value and the actual timestamp in ALLOC messages differ by a couple of seconds. This behavior occurs because the syslog mechanism contains a slight difference when it reads the time (as seen in PORT_BLOCK_ALLOC syslog) and the time at which NAT application reads the time (to update the ALLOC time in the subscriber context). The interim system log displays the ALLOC time retrieved from the subscriber context.

- Because these logs are generated on CPU computation and in the fast path, a slight impact might be observed with fast path performance only when a generation of the log occurs.

- Port block creation timestamp in hexadecimal is saved in the JSERVICES_NAT_PORT_BLOCK_RELEASE message, even if interim logging is not present.

- If you define the logging interval when traffic flow is in progress, this functionality takes effect on existing and new flows. You need not reboot the MIC or activate and deactivate the service set.

- If the flows or subscribers are timing out, it denotes that no new packets or traffic flows are seen for this 5-tuple data or for that particular subscriber. In such a case, interim logs are not generated.

- If the interim-logging interval is lower than the inactivity-timeout of the flow, interim logs are not observed when the flow is timing out and the interim-logging interval has elapsed. If the interim-logging interval is lower than the subscriber-timeout value, interim logs are not observed when the subscriber is timing out and the interim-logging interval has elapsed. For example, if the inactivity-timeout is configured to 2500 seconds and the interim-logging is configured as 1800 seconds, when the flow is timing out, there is a point in time when 1800 seconds has elapsed since the last packet was seen on this flow and no interim log is generated in this case.

- The interim logs are recorded for those pools that have PBA configured. If pools exist without the PBA configuration present on the service network processing unit (NPU), interim logs are not saved even if you enable the interim logging functionality.

- You can configure only a range of values for the interval at which the logs need to be generated, such as 0, [1800, 86400].

- You can enable the generation of syslogs by using the *syslog* statement at the `[edit system]` and `[edit services service-sets` *`service-set name`* `nat rule` *`rule-name`* `term` *`term-name`* `then]` hierarchy levels that contain the NAT rules with PBA pools. Interim logs are not triggered if the recording of syslogs are not enabled on the system.

- We recommend that you configure the interim-logging interval to be higher than the inactivity timeout period for established flows. Also, we recommend that you configure the interim-logging interval to be higher than the subscriber-timeout value. When endpoint-independent mapping (EIM) is configured, the interim-logging interval must be higher than the sum of the address pooling paired (APP) timeout and EIM timeout values.

- Transmission of logs occurs in clear-text format similar to other log messages that the services PICs do not encrypt. It is assumed that the transport of logs and the positioning of the log collector are within a secured realm. Because the messages do not contain sensitive details such as username or passwords, the messages do not cause any security or reliability risks. Increased generation of log messages does not cause a possibility of a flood of logs because the frequency of logging can be configured, depending on the network topology, traffic levels, and your monitoring needs.

- The logs for PBA in the microkernel start with the prefix of ASP_*. These logs have been modified to start with the prefix of JSERVICES_*. The following are examples of system logs for PBA in the microkernel and with the Junos OS Extension-Provider packages installed and configured on the device.

  ```
  Microkernel: 1970-01-01 00:32:36 {nat64}[FWNAT]:ASP_NAT_PORT_BLOCK_ACTIVE: 2001:db8:0:0:0:0:0:2 ->
  10.1.1.1:1050-1091 0x6f
  ```

  ```
  Junos OS Extension-Provider (eJunos): 1970-01-01 00:32:36 {nat64}[FWNAT]:JSERVICES_NAT_PORT_BLOCK_ACTIVE:
  2001:db8:0:0:0:0:0:2 -> 10.1.1.1:1050-1091 0x6f
  ```

- Also, you can specify the interim logging interval per NAT pool instead of a global configuration per MS-PIC, based on whether you want the syslog settings to apply to all the NAT pools on a device or for a particular NAT pool. For NAT, the member interfaces must have the jservices-nat package configured. The JSERVICES_NAT_PORT_BLOCK_ACTIVE system logging message is generated when you configure interim logging for PBA. The following sample logs denote the log messages generated with the interim interval set as 1800 seconds. You can notice that the timestamp between consecutive interim logs is more than 1800 seconds.

  ```
  1970-01-01 00:01:51 [FWNAT]:JSERVICES_NAT_PORT_BLOCK_ALLOC: 2001:db8:0:0:0:0:0:2 ->
  10.1.1.1:1050-1091
  1970-01-01 00:32:36 {nat64}[FWNAT]:JSERVICES_NAT_PORT_BLOCK_ACTIVE: 2001:db8:0:0:0:0:0:2 ->
  10.1.1.1:1050-1091 0x6f
  1970-01-01 01:03:20 {nat64}[FWNAT]:JSERVICES_NAT_PORT_BLOCK_ACTIVE: 2001:db8:0:0:0:0:0:2 ->
  10.1.1.1:1050-1091 0x6f
  1970-01-01 01:34:04 {nat64}[FWNAT]:JSERVICES_NAT_PORT_BLOCK_ACTIVE: 2001:db8:0:0:0:0:0:2 ->
  10.1.1.1:1050-1091 0x6f
  1970-01-01 02:04:48 {nat64}[FWNAT]:JSERVICES_NAT_PORT_BLOCK_ACTIVE: 2001:db8:0:0:0:0:0:2 ->
  10.1.1.1:1050-1091 0x6f
  ```

- Starting in Junos OS release 19.3R1, when you configure a softwire prefix other than 128, all the JSERVICES_NAT_PORT_BLOCK logs now displays the prefixed B4 address. The following JSERVICES_NAT_PORT_BLOCK are modified:

  - JSERVICES_NAT_PORT_BLOCK_ALLOC

  - JSERVICES_NAT_PORT_BLOCK_RELEASE

- JSERVICES_NAT_PORT_BLOCK_ACTIVE

In earlier Junos OS releases, when a softwire prefix was configured, some of the B4 addresses displayed in the JSERVICES_NAT_PORT_BLOCK log were /128 addresses. For example, when a /56 prefix was configured, the port block syslog displayed the following B4 addresses:

- The JSERVICES_NAT_PORT_BLOCK_ALLOC displayed the /128 B4 address of the first B4 which was allocated a port from a particular port block
- The JSERVICES_NAT_PORT_BLOCK_RELEASE displayed the /128 B4 address of the last B4 which released its port back to the port block

SEE ALSO

Configuring NAT Session Logs | 347

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
15.1R1	Starting in Junos OS Release 15.1R1, you can also configure the interim logging interval at a NAT pool level.

CHAPTER 13

Port Forwarding

IN THIS CHAPTER

- Port Forwarding | 276

Port Forwarding

IN THIS SECTION

- Port Forwarding Overview | 276
- Configuring Port Forwarding for Static Destination Address Translation | 277
- Configuring Port Forwarding Without Destination Address Translation | 281
- Example: Configuring Port Forwarding with Twice NAT | 284

Port Forwarding Overview

IN THIS SECTION

- Benefits of Port Forwarding | 277

You can map an external IP address and port with an IP address and port in a private network. This mapping, called port forwarding, is supported on the MS-DPC, MS-100, MS-400, and MS-500 MultiServices PICS. Starting in Junos OS Release 17.4R1, port forwarding is also supported on the MS-MPC and MS-MIC.

Port forwarding allows the destination address and port of a packet to be changed to reach the correct host in a Network Address Translation (NAT) gateway. The translation facilitates reaching a host within a masqueraded, typically private, network, based on the port number on which the packet was received from the originating host. An example of this type of destination is the host of a public HTTP server within a private network. You can also configure port forwarding without translating a destination address. Port forwarding supports endpoint-independent mapping (EIM), endpoint-independent filltering (EIF), and address pooling paired (APP).

Port forwarding works only with the FTP application-level gateway (ALG), and has no support for technologies that offer IPv6 services over IPv4 infrastructure, such as IPv6 rapid deployment (6rd) and dual-stack lite (DS-Lite). Port forwarding supports only `dnat-44` and `twice-napt-44` on IPv4 networks.

Benefits of Port Forwarding

- Allows remote computers, such as public machines on the Internet, to connect to a non-standard port of a specific computer that is hidden within a private network.

Configuring Port Forwarding for Static Destination Address Translation

You can configure destination address translation with port forwarding. Port forwarding allows the destination address and port of a packet to be changed to reach the correct host in a Network Address Translation (NAT) gateway. Port forwarding is supported on the MS-DPC, MS-100, MS-400, and MS-500 MultiServices PICS. Starting in Junos OS Release 17.4R1, port forwarding is also supported on the MS-MPC and MS-MIC.

To configure destination address translation with port forwarding:

1. In configuration mode, go to the `[edit services nat]` hierarchy level.

   ```
   [edit]
   user@host# edit services nat
   ```

2. Configure the NAT pool with an address.

   ```
    [edit services nat]
   user@host# set pool pool-name address address
   ```

   In the following example, `dest-pool` is used as the pool name and `192.0.2.2` as the address.

   ```
   user@host# set pool dest-pool address 192.0.2.2
   ```

3. Configure the rule, match direction, term, and destination address.

```
[edit services nat]
user@host# set rule rule-name match-direction match-direction term term-name from
destination-address  address
```

In the following example, the name of the rule is `rule-dnat44`, the match direction is `input`, the name of the term is `t1`, and the address is `198.51.100.20`.

```
[edit services nat]
user@host# set rule rule-dnat44 match-direction  input term t1 from destination-address
198.51.100.20
```

4. Configure the destination port range.

```
[edit services nat]
user@host# set rule rule-name match-direction match-direction term term-name from
destination-port range high maximum-value low minimum-value
```

In the following example, the upper port range is `50` and the lower port range is `20`.

```
[edit services nat]
user@host# set rule rule-dnat44 match-direction  input term t1 from destination-port range
high 50 low 20
```

5. Go to the [edit services nat rule *rule-name* term *term-name*] hierarchy level.

```
[edit services nat]
user@host# edit rule rule-name term term-name
```

6. Configure the destination pool.

```
[edit services nat rule rule-name term term-name]
user@host# set then translated destination-pool dest-pool-name
```

In the following example, the destination pool name is `dest-pool`.

```
[edit services nat rule rule-dnat44 term t1]
user@host# set then translated destination-pool  dest-pool
```

7. Specify the name of the mapping for port forwarding and configure the translation type. You can only configure one mapping within a NAT rule term.

```
[edit services nat rule rule-name term term-name]
user@host# set then port-forwarding-mappings  map-name
user@host# set then translated translation-type translation-type
```

In the following example, the port forwarding mapping name is `map1`, and the translation type is `dnat-44`.

```
[edit services nat rule rule-dnat44 term t1]
user@host# set then port-forwarding-mappings  map1
user@host# set then translated translation-type dnat-44
```

8. Go to the `[edit services nat port-forwarding map-name]` hierarchy level.

```
[edit services nat]
user@host# edit port-forwarding map-name
```

9. Configure the mapping for port forwarding.

```
[edit port-forwarding map-name]
user@host# set destined-port port-id
user@host# set translated-port port-id
```

In the following example, the destination port number that needs to be translated is `23` and the port to which traffic is mapped is `45`.

```
[edit port-forwarding map1]
user@host# set destined-port 23
user@host# set translated-port 45
```

NOTE:

- Multiple port mappings are supported with port forwarding. Up to 32 port maps can be configured for port forwarding.
- The destination port should not overlap the port range configured for NAT.

10. Apply the NAT rule to the service set that performs the port mapping.

```
[edit services service-set service-set-name]
user@host# set nat-rules rule-name
```

11. Verify the configuration by using the `show` command at the `[edit services nat]` hierarchy level.

```
[edit services]
user@host# show
nat {
    pool dest-pool {
        address 192.0.2.2/32;
    }
    rule rule-dnat44 {
        match-direction input;
        term t1
          from {
             destination-address {
                   198.51.100.20/32
             }
             destination-port {
                   range low 20 high 50;
               }
        }
          then {
             port-forwarding-mappings map1;
             translated {
                  destination-pool dest-pool;
                  translation-type {
                     dnat-44;
                             }
                        }
                 }
       }
    }
    port-forwarding map1 {
       destined-port 45;
       translated-port 23;
        }
}
service-set ss1 {
     nat-rules rule-dnat44;
     interface-service {
        service-interface sp-10/0/0.0;
        }
     }
```

NOTE:

- A similar configuration is possible with twice NAT for IPv4. See "Example: Configuring Port Forwarding with Twice NAT" on page 284 .

- Port forwarding and stateful firewall can be configured together. Stateful firewall has precedence over port forwarding.

Configuring Port Forwarding Without Destination Address Translation

You can configure port forwarding without translating a destination address. Port forwarding allows the destination port to be changed to reach the correct port in a Network Address Translation (NAT) gateway. Port forwarding is supported on the MS-DPC, MS-100, MS-400, and MS-500 MultiServices PICS. Starting in Junos OS Release 17.4R1, port forwarding is also supported on the MS-MPC and MS-MIC.

To configure port forwarding without destination address translation in IPv4 networks:

1. In configuration mode, go to the [edit services nat] hierarchy level.

```
[edit]
user@host# edit services nat
```

2. Configure the rule, match direction, term name, and any conditions that the traffic must match before the rule is applied.

   ```
   [edit services nat]
   user@host# set rule rule-name match-direction match-direction term term-name from match-
   conditions
   ```

   In the following example, the name of the rule is `rule-port-forwarding`, the match direction is `input`, the name of the term is `t1`, and the destination address that must be matched is `198.51.100.20`.

   ```
   [edit services nat]
   user@host# set rule rule-port-forwarding match-direction  input term t1 from destination-
   address 198.51.100.20
   ```

3. Go to the `[edit services nat rule` *`rule-name`* `term` *`term-name`*`]` hierarchy level.

   ```
   [edit services nat]
   user@host# edit rule  rule-name term term-name
   ```

4. Specify that there is no address translation for this rule.

   ```
   [edit services nat rule rule-name term term-name]
   user@host# set then no-translation
   ```

5. Specify the name of the mapping for port forwarding. You can only configure one mapping within a NAT rule term.

   ```
   [edit services nat rule rule-name term term-name]
   user@host# set then port-forwarding-mappings map-name
   ```

   In the following example, the port forwarding mapping name is `map1`.

   ```
   [edit services nat rule rule-port-forwarding term t1]
   user@host# set then port-forwarding-mappings  map1
   ```

6. Go to the [edit services nat port-forwarding *map-name*] hierarchy level.

```
[edit services nat]
user@host# edit port-forwarding map-name
```

7. Configure the mapping for port forwarding.

```
[edit port-forwarding map-name]
user@host# set destined-port port-id
user@host# set translated-port port-id
```

In the following example, the destination port number that needs to be translated is 23 and the port to which traffic is mapped is 45.

```
[edit port-forwarding map1]
user@host# set destined-port 23
user@host# set translated-port 45
```

NOTE:

- Multiple port mappings are supported with port forwarding. Up to 32 port maps can be configured for port forwarding.

- The destination port should not overlap the port range configured for NAPT.

8. Apply the NAT rule to the service set that performs the port mapping.

```
[edit services service-set service-set-name]
user@host# set nat-rules rule-name
```

NOTE: On the MS-MPC and MS-MIC, you cannot apply port forwarding NAT rules to an AMS interface.

9. Verify the configuration by using the show command at the [edit services] hierarchy level.

```
[edit services]
user@host# show
```

```
nat {
    rule rule-port-forwarding {
        match-direction input;
        term t1 {
            then {
               port-forwarding-mappings map1;
                        no-translation                         }
                    }
           }
    }
    port-forwarding map1 {
        destined-port 45;
       translated-port 23;
        }
}
service-set ss2 {
     nat-rules rule-port-forwarding;
     interface-service {
        service-interface sp-10/0/0.0;
        }
     }
```

NOTE: Port forwarding and stateful firewall can be configured together. Stateful firewall has precedence over port forwarding.

Example: Configuring Port Forwarding with Twice NAT

The following example configures port forwarding with `twice-napt-44` as the translation type. The example also has stateful firewall and multiple port maps configured.

Port forwarding is supported on the MS-DPC, MS-100, MS-400, and MS-500 MultiServices PICS. Starting in Junos OS Release 17.4R1, port forwarding is also supported on the MS-MPC and MS-MIC.

```
[edit services]
user@host# show
service-set in {
    syslog {
        host local {
            services any;
        }
    }
    stateful-firewall-rules r;
    nat-rules r;
    interface-service {
        service-interface sp-10/0/0.0;
    }
}
stateful-firewall {
    rule r {
        match-direction input;
        term t {
            from {
                destination-port {
                    range low 20 high 5000;
                }
            }
            then {
                reject;
            }
        }
    }
}
nat {
    pool x {
        address 203.0.113.2/32;
    }
    rule r {
        match-direction input;
        term t {
            from {
                destination-address {
                    198.51.100.2/32;
                }
                destination-port {
                    range low 10 high 20000;
                }
            }
            then {
                port-forwarding-mappings y;
                translated {
                   destination-pool x;
                   translation-type {
                      twice-napt-44;
                    }
                }
            }
        }
    }
    port-forwarding y {
        destined-port 45;
        translated-port 23;
        destined-port 55;
        translated-port 33;
        destined-port 65;
        translated-port 43;
         }
}
adaptive-services-pics {
    traceoptions {
        file sp-trace;
        flag all;
    }
}
```

NOTE:

- Stateful firewall has precedence over port forwarding. In this example, for instance, no traffic destined to any port between 20 and 5000 will be translated.
- Up to 32 port maps can be configured.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
17.4R1	Starting in Junos OS Release 17.4R1, port forwarding is also supported on the MS-MPC and MS-MIC.
17.4R1	Starting in Junos OS Release 17.4R1, port forwarding is also supported on the MS-MPC and MS-MIC.
17.4R1	Starting in Junos OS Release 17.4R1, port forwarding is also supported on the MS-MPC and MS-MIC.
17.4R1	Starting in Junos OS Release 17.4R1, port forwarding is also supported on the MS-MPC and MS-MIC.

CHAPTER 14

Dynamic Address-Only Source Translation

IN THIS CHAPTER

- Dynamic Address-Only Source Translation | 287

Dynamic Address-Only Source Translation

IN THIS SECTION

- Configuring Dynamic Address-Only Source Translation in IPv4 Networks | 287
- Example: Dynamic Source NAT as a Next-Hop Service | 294
- Example: Assigning Addresses from a Dynamic Pool for Static Use | 296

Configuring Dynamic Address-Only Source Translation in IPv4 Networks

In IPv4 networks, dynamic address translation (dynamic NAT) is a mechanism to dynamically translate the destination traffic without port mapping. To use dynamic NAT, you must specify a source pool name, which includes an address configuration.

To configure dynamic NAT in IPv4 networks:

1. In configuration mode, go to the `[edit services]` hierarchy level.

```
[edit]
user@host# edit services
```

2. Configure the service set and NAT rule.

```
[edit services]
user@host# set service-set service-set-name nat-rules rule-name
```

In the following example, the name of the service set is **s1**, and the name of the NAT rule is **rule-dynamic-nat44**.

```
[edit services]
user@host# set service-set s1 nat-rules rule-dynamic-nat44
```

3. Go to the `[interface-service]` hierarchy level for the service set.

```
[edit services]
user@host# edit service-set s1 interface-service
```

4. Configure the service interface.

```
 [edit services service-set s1 interface-service]
user@host# set service-interface service-interface-name
```

In the following example, the name of the service interface is **ms-0/1/0**.

> **NOTE**: If the service interface is not present in the router, or the specified interface is not functional, the following command can result in an error.

```
 [edit services service-set s1 interface-service]
user@host# set service-interface ms-0/1/0
```

5. Go to the `[edit services nat]` hierarchy level. Issue the following command from the top of the services hierarchy, or use the **top** keyword.

```
 [edit services service-set s1 interface-service]
user@host# top editservices nat
```

6. Configure the NAT pool with an address.

```
[edit services nat]
user@host# set pool pool-name address address
```

In the following example, the name of the pool is **source-dynamic-pool**, and the address is **10.10.10.0**.

```
[edit services nat]
user@host# set pool source-dynamic-pool address 10.10.10.0
```

7. Configure the rule, match direction, term, and source address.

```
[edit services nat]
user@host# set rule rule-name match-direction match-direction term term-name from source-
address address
```

In the following example, the name of the rule is **rule-dynamic-nat44**, the match direction is **input**, the name of the term is **t1**, and the source address is **3.1.1.0**.

```
[edit services nat]
user@host# set rule rule-dynamic-nat44 match-direction  input term t1 from source-address
3.1.1.0
```

8. Go to the `[edit rule rule-dynamic-nat-44 term t1]` hierarchy level.

```
[edit services nat]
user@host# edit rule rule-dynamic-nat44 term t1
```

9. Configure the source pool and the translation type.

```
[edit services nat rule rule-dynamic-nat44 term t1]
user@host# set then translated source-pool src-pool-name translation-type translation-type
```

In the following example, the name of the source pool is **source-dynamic-pool** and the translation type is **dynamic-nat44**.

```
[edit services nat rule rule-dynamic-nat44 term t1]
user@host# set then translated source-pool  source-dynamic-pool translation-type dynamic-
nat44
```

10. Go to the `[edit services adaptive-services-pics]` hierarchy level. In the following command, the **top** keyword ensures that the command is run from the top of the hierarchy.

```
[edit services nat rule rule-dynamic-nat44 term t1]
user@host# top editservices adaptive-services-pics
```

11. Configure the trace options.

```
[edit services adaptive-services-pics]
user@host# set traceoptions flag tracing parameter
```

In the following example, the tracing parameter is configured as **all**.

```
[edit services adaptive-services-pics]
user@host# set traceoptions flag all
```

12. Verify the configuration by using the `show` command at the `[edit services]` hierarchy level.

```
[edit services]
user@host# show
service-set s1 {
    nat-rules rule-dynamic-nat44;
    interface-service {
        service-interface ms-0/1/0;
    }
}
 nat {
    pool source-dynamic-pool {
        address 10.1.1.0/24;
    }
    rule rule-dynamic-nat44 {
        match-direction input;
        term t1 {
            from {
                source-address {
                    3.1.1.0/24;
                }
            }
            then {
                translated {
                    destination-pool source-dynamic-pool;
                    translation-type {
                        dynamic-nat44;
                    }
                }
            }
        }
    }
}
 adaptive-services-pics {
    traceoptions {
        flag all;
    }
  }
```

The following example configures the translation type as **dynamic-nat44**.

```
[edit services]
user@host# show
service-set s1 {
    nat-rules rule-dynamic-nat44;
    interface-service {
        service-interface ms-0/1/0;
    }
}
 nat {
    pool source-dynamic-pool {
        address 10.1.1.0/24;
    }
    rule rule-dynamic-nat44 {
        match-direction input;
        term t1 {
            from {
                source-address {
                    3.1.1.0/24;
                }
            }
            then {
                translated {
                    destination-pool source-dynamic-pool;
                    translation-type {
                        dynamic-nat44;
                    }
                }
            }
        }
    }
}
 adaptive-services-pics {
    traceoptions {
        flag all;
    }
  }
```

The following configuration specifies that NAT is not performed on incoming traffic from the source address **192.168.20.24/32** by providing a NAT rule term `t0` that configures `no-translation`. Dynamic NAT is performed on all other incoming traffic, as configured by term `t1` of the NAT rule. The `no-translation` option is supported on MX Series routers with MS-DPCs and on M Series routers with MS-100, MS-400, and MS-500 MultiServices PICS. The `no-translation` option is supported on MX Series routers with MS-MPCs and MS-MICs starting in Junos OS release 15.1R1.

```
[edit services nat]
pool my-pool {
    address-range low 10.10.10.1 high 10.10.10.16;
    port automatic;
}
rule src-nat {
    match-direction input;
    term t0 {
        from {
            source-address 192.168.20.24/32;
        }
        then {
            no-translation;
        }
    }
    term t1 {
        then {
            translated {
                translation-type dynamic-nat44;
                source-pool my-pool;
            }
        }
    }
}
```

The following configuration performs NAT using the source prefix **20.20.10.0/24** without defining a pool.

```
[edit services nat]
rule src-nat {
    match-direction input;
    term t1 {
        then {
            translation-type dynamic-nat44;
            source-prefix 20.20.10.0/24;
        }
    }
}
```

The following configuration performs NAT using the destination prefix **20.20.10.0/32** without defining a pool.

```
[edit services nat]
rule src-nat {
    match-direction input;
    term t1 {
        from {
            destination-address 10.10.10.10/32;
            then {
                translation-type dnat44;
                destination-prefix 20.20.10.0/24;
            }
        }
    }
}
```

Example: Dynamic Source NAT as a Next-Hop Service

The following example shows dynamic-source NAT applied as a next-hop service:

```
[edit interfaces]
ge-0/2/0 {
    unit 0 {
        family mpls;
    }
}
sp-1/3/0 {
    unit 0 {
        family inet;
    }
    unit 20 {
        family inet;
    }
    unit 32 {
        family inet;
    }
}
[edit routing-instances]
protected-domain {
    interface ge-0/2/0.0;
    interface sp-1/3/0.20;
    instance-type vrf;
    route-distinguisher 10.58.255.17:37;
    vrf-import protected-domain-policy;
    vrf-export protected-domain-policy;
    routing-options {
        static {
            route 0.0.0.0/0 next-hop sp-1/3/0.20;
        }
    }
}
[edit policy-options]
policy-statement protected-domain-policy {
    term t1 {
        then reject;
    }
}
[edit services]
```

```
stateful-firewall {
    rule allow-all {
        match-direction input;
        term t1 {
            then {
                accept;
            }
        }
    }
}
nat {
    pool my-pool {
        address 10.58.16.100;
        port automatic;
    }
    rule hide-all {
        match-direction input;
        term t1 {
            then {
                translated {
                    source-pool my-pool;
                    translation-type napt-44;
                }
            }
        }
    }
}
service-set null-sfw-with-nat {
    stateful-firewall-rules allow-all;
    nat-rules hide-all;
    next-hop-service {
        inside-service-interface sp-1/3/0.20;
        outside-service-interface sp-1/3/0.32;
    }
}
```

Example: Assigning Addresses from a Dynamic Pool for Static Use

The following configuration statically assigns a subset of addresses that are configured as part of a dynamic pool (`dynamic-pool`) to two separate static pools (`static-pool` and `static-pool2`).

```
[edit services nat]
pool dynamic-pool {
    address 20.20.10.0/24;
}
pool static-pool {
    address-range low 20.20.10.10 high 10.20.10.12;
}
pool static-pool2 {
    address 20.20.10.15/32;
}
rule src-nat {
    match-direction input;
    term t1 {
        from {
            source-address 30.30.30.0/24;
        }
        then {
            translation-type dynamic-nat44;
            source-pool dynamic-pool;
        }
    }
    term t2 {
        from {
            source-address 10.10.10.2;
        }
        then {
            translation-type basic-nat44;
            source-pool static-pool;
        }
    }
    term t3 {
        from {
            source-address 10.10.10.10;
        }
        then {
            translation-type basic-nat44;
            source-pool static-pool2;
        }
    }
}
```

CHAPTER 15

Inline NAT

IN THIS CHAPTER

- Inline NAT | 298

Inline NAT

IN THIS SECTION

- Inline Network Address Translation Overview | 298
- Example: Configuring Inline Network Address Translation—Interface-Based Method | 300
- Example: Configuring Inline Network Address Translation—Route-Based Method | 309

Inline Network Address Translation Overview

IN THIS SECTION

- Benefits of Inline NAT | 300

Inline NAT uses the capabilities of the MPC line card, eliminating the need for a services card for NAT. Consequently, you can achieve line-rate, low-latency address translations (up to 120 Gbps per slot). The current implementation provides:

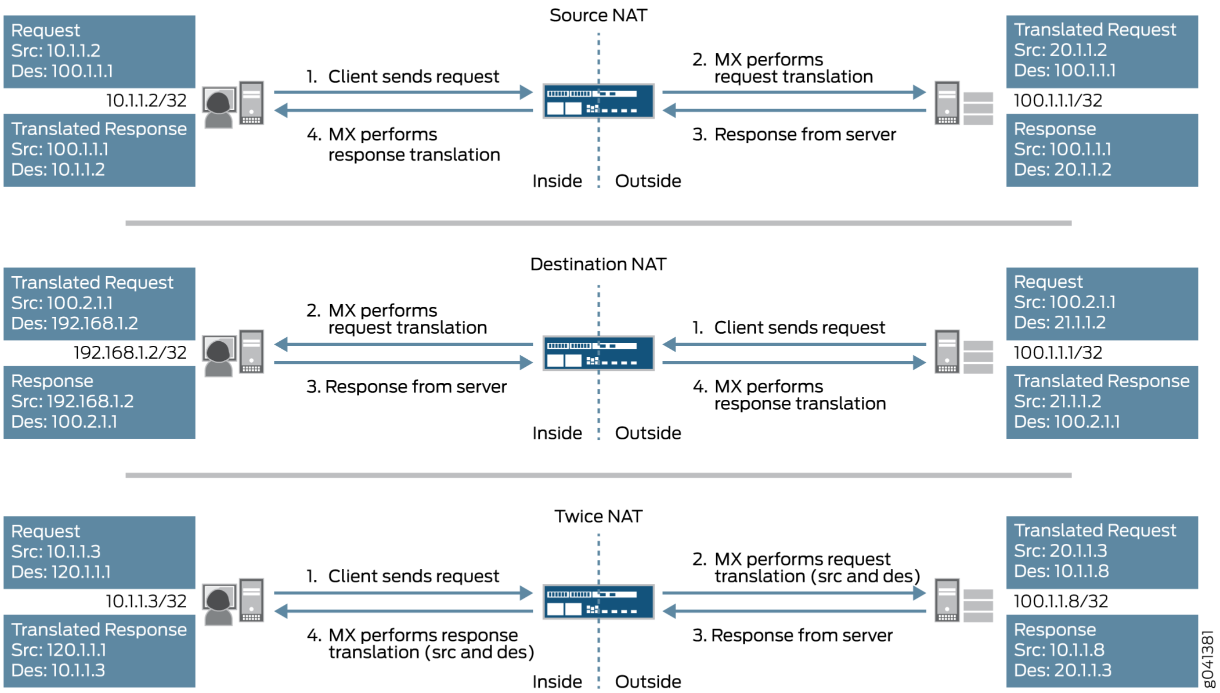
- 1:1 static address mapping.
- Bidirectional mapping - source NAT for outbound traffic and destination NAT for inbound traffic.

- No limit on number of flows.

- Support for Source, destination, and twice NAT, as shown in Figure 16 on page 299 . Inline NAT supports the translation type `basic-nat44`. Starting in Junos OS Release 15.1R1, inline NAT also supports `twice-basic-nat-44`.

- Support for hairpinning.

Figure 16: Supported Inline NAT Types



To configure inline NAT, you define your service interface as type `si-` (service-inline) interface. You must also reserve adequate bandwidth for the inline interface. This enables you to configure both interface or next-hop service-sets used for NAT. The `si-` interface serves as a “virtual service PIC”.

NOTE:

- Only static NAT is supported. Port translation, dynamic NAT, and ALGs are not supported. Hence, applications such as SIP or FTP Active Mode which require advanced processing for NAT do not function. An MS-MPC, MS-MIC, MS-DPC, or MS-PIC is still needed for any stateful-firewall processing, ALG support, and dynamic port translation.

- Inline NAT does not support sampling or logging of packets.

Benefits of Inline NAT

- Eliminates the need for a services card
- Supports more NAT flows than a services card

SEE ALSO

Network Address Translation Configuration Overview | 92

Carrier-Grade NAT Feature Comparison for Junos Address Aware by Type of Interface Card | 64

Example: Configuring Inline Network Address Translation—Interface-Based Method

IN THIS SECTION

- Requirements | 300
- Overview and Topology | 301
- Configuration | 302
- Verification | 307

This configuration example illustrates how to configure interface-based inline network address translation (NAT) on MX Series devices using `si-` (service-inline) interfaces with interface-style service-sets.

This topic covers:

Requirements

This example uses the following hardware and software components:

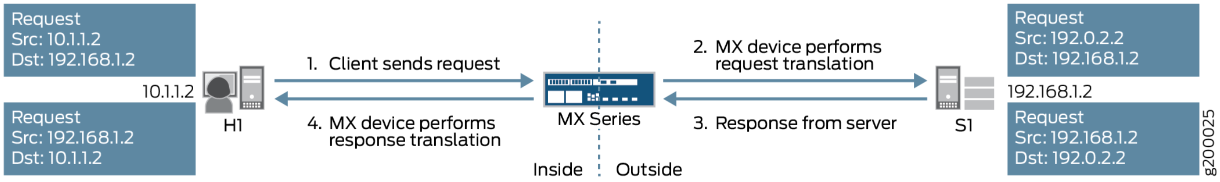
- MX Series router with a Modular Port Concentrator (MPC) line card
- Junos OS Release 11.4R1 or higher

Overview and Topology

As of Junos OS Release 11.4R1, MPC line cards can perform some services without the need of a dedicated services card, such as an MS-MPC. Inline services generally provide better performance than using a services card, however their functionality tends to be more basic. For example, inline NAT supports only static NAT.

In this example, an MX Series device with an MPC line card provides inline source NAT services to traffic flowing between two end hosts. The topology for this scenario is shown in Figure 17 on page 301

Figure 17: Inline Source NAT Using an MX Series Device with an MPC



As shown in the figure, host H1 sends traffic towards server S1. The MX Series device performs source NAT to translate H1’s source IP address from 10.1.1.2 to 192.0.2.2. Server S1 then sends return traffic to host H1 using the destination IP address 192.0.2.2, and the MX Series device reverts H1’s IP address back to 10.1.1.2.

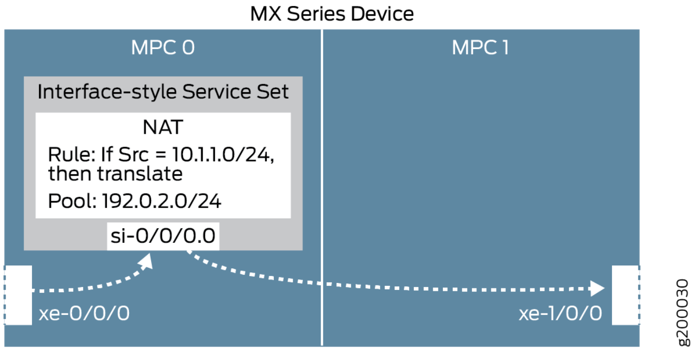
The following configuration elements are used in this scenario:

- Inline service interface—a virtual interface that resides on the Packet Forwarding Engine of the MPC. To access services, traffic flows in and out of these `si-` (service-inline) interfaces.
- Service set—defines the service(s) to be performed, and identifies which inline interface(s) will feed traffic into and out of the service set. There are two ways to implement service sets:
  - Interface-style—an interface-based method, where packets arriving at an interface are forwarded through the inline service.
  - Next-hop-style—a route-based method, where static routes are used to forward packets destined for a specific destination through the inline service.

  This example uses the interface-style service set.
- NAT rule—uses an if-then structure (similar to firewall filters) to define matching conditions and then apply address translation to the matching traffic.
- NAT pool—a user-defined set of IP addresses that are used by the NAT rule for translation.

These elements come together as shown in Figure 18 on page 302

Figure 18: Interface-Based Inline Source NAT



Configuration

IN THIS SECTION

- CLI Quick Configuration | 302
- Enable Inline Services and Create an Inline Interface | 303
- Configure NAT Rule and Pool | 304
- Configure the (Interface-style) Service Set | 304
- Configure Physical Interfaces | 305

To configure inline NAT using an interface-style service set, perform these tasks:

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the [edit] hierarchy level.

```
## Enable inline services, create an si- interface, reserve bandwidth ##
set chassis fpc 0 pic 0 inline-services bandwidth 1g
set interfaces si-0/0/0 unit 0 family inet
## Configure a NAT rule and pool ##
set services nat rule SRC-NAT1 match-direction input
set services nat rule SRC-NAT1 term r1 from source-address 10.1.1.0/24
set services nat rule SRC-NAT1 term r1 then translated translation-type basic-nat44
set services nat rule SRC-NAT1 term r1 then translated source-pool p1
set services nat pool p1 address 192.0.2.0/24
## Configure the (interface-style) service set ##
set services service-set INT-STYLE-SS-NAT1 nat-rules SRC-NAT1
set services service-set INT-STYLE-SS-NAT1 interface-service service-interface si-0/0/0.0
## Configure interfaces ##
set interfaces xe-0/0/0 unit 0 family inet address 10.1.1.1/24
set interfaces xe-0/0/0 description INSIDE
set interfaces xe-1/0/0 unit 0 family inet address 192.168.1.1/24
set interfaces xe-1/0/0 description OUTSIDE
set interfaces xe-0/0/0 unit 0 family inet service input service-set INT-STYLE-SS-NAT1
set interfaces xe-0/0/0 unit 0 family inet service output service-set  INT-STYLE-SS-NAT1
```

Enable Inline Services and Create an Inline Interface

Step-by-Step Procedure

1. Enable inline services for the relevant FPC slot and PIC slot, and define the amount of bandwidth to dedicate for inline services.

   The FPC and PIC settings here will create and map to an `si-` interface.

   ```
   [edit chassis fpc 0 pic 0]
   user@MX# set inline-services bandwidth 1g
   ```

2. On the `si-` interface, specify the protocol family (or families) that will need NAT services.

   NOTE: The FPC and PIC settings here must match the settings defined above.

   ```
   [edit interfaces si-0/0/0]
   user@MX# set unit 0 family inet
   ```

Configure NAT Rule and Pool

Step-by-Step Procedure

1. Configure a NAT rule that matches on traffic arriving at the MX device from H1's subnet (10.1.1.0/24), translates it using basic IPv4 NAT, and uses an IP address from pool `p1`.

   ```
   [edit services nat]
   user@MX# set rule SRC-NAT1 match-direction input
   user@MX# set rule SRC-NAT1 term r1 from source-address 10.1.1.0/24
   user@MX# set rule SRC-NAT1 term r1 then translated translation-type basic-nat44
   user@MX# set rule SRC-NAT1 term r1 then translated source-pool p1
   ```

2. Configure the NAT pool.

   ```
   [edit services nat]
   user@MX# set pool p1 address 192.0.2.0/24
   ```

Configure the (Interface-style) Service Set

Step-by-Step Procedure

1. Configure a service set that uses the inline NAT service (`nat-rules`), and the inline interface defined above. Use the `interface-service` parameter to specify that this is an interface-style service set.

   Traffic will flow into and out of the `si-` interface to access the inline NAT service.

   ```
   [edit services]
   user@MX# set service-set INT-STYLE-SS-NAT1 nat-rules SRC-NAT1
   user@MX# set service-set INT-STYLE-SS-NAT1 interface-service service-interface si-0/0/0.0
   ```

Configure Physical Interfaces

Step-by-Step Procedure

1. Configure the physical interfaces.

```
[edit interfaces]
user@MX# set xe-0/0/0 unit 0 family inet address 10.1.1.1/24
user@MX# set xe-0/0/0 description INSIDE
user@MX# set xe-1/0/0 unit 0 family inet address 192.168.1.1/24
user@MX# set xe-1/0/0 description OUTSIDE
```

2. On the ’inside’ interface, specify that traffic will be sent through the service set defined above.

```
[edit interfaces xe-0/0/0 unit 0]
user@MX# set family inet service input service-set INT-STYLE-SS-NAT1
user@MX# set family inet service output service-set  INT-STYLE-SS-NAT1
```

Results

```
chassis {
    fpc 0 {
        pic 0 {
            inline-services {
                bandwidth 1g;
            }
        }
    }
}

services {
    service-set INT-STYLE-SS-NAT1 {
        nat-rules SRC-NAT1;
        interface-service {
            service-interface si-0/0/0.0;
        }
    }
    nat {
        pool p1 {
            address 192.0.2.0/24;
        }
        rule SRC-NAT1 {
            match-direction input;
            term r1 {
                from {
                    source-address {
                        10.1.1.0/24;
                    }
                }
                then {
                    translated {
                        source-pool p1;
                        translation-type {
                            basic-nat44;
                        }
                    }
                }
            }
        }
    }
}

interfaces {
    si-0/0/0 {
        unit 0 {
            family inet;
        }
    }
    xe-0/0/0 {
        description INSIDE;
        unit 0 {
            family inet {
                service {
                    input {
                        service-set INT-STYLE-SS-NAT1;
                    }
                    output {
                        service-set INT-STYLE-SS-NAT1;
                    }
                }
                address 10.1.1.1/24;
            }
        }
    }
    xe-1/0/0 {
        description OUTSIDE;
        unit 0 {
            family inet {
                address 192.168.1.1/24;
            }
        }
    }
}
```

Verification

IN THIS SECTION

- Verifying Reachability from Host H1 to Server S1 | 307
- Verifying Address Translation | 308

Confirm that the configuration is working properly.

Verifying Reachability from Host H1 to Server S1

Purpose

Verify reachability between H1 and S1.

Action

On host H1, verify that the host can ping server S1.

```
user@H1> ping 192.168.1.2 count 5
PING 192.168.1.2 (192.168.1.2): 56 data bytes
64 bytes from 192.168.1.2: icmp_seq=0 ttl=63 time=0.991 ms
64 bytes from 192.168.1.2: icmp_seq=1 ttl=63 time=14.186 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=63 time=3.016 ms
64 bytes from 192.168.1.2: icmp_seq=3 ttl=63 time=3.742 ms
```

```
64 bytes from 192.168.1.2: icmp_seq=4 ttl=63 time=4.748 ms

--- 192.168.1.2 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.991/5.337/14.186/4.593 ms
```

Meaning

H1 can successfully reach S1.

Verifying Address Translation

Purpose

Verify that address translation is working correctly.

Action

1. On the MX device, verify that the inline NAT configuration details have been applied correctly.

   ```
   user@MX> show services inline nat pool
   Interface: si-0/0/0, Service set: INT-STYLE-SS-NAT1
     NAT pool: p1, Translation type: BASIC NAT44
       Address range: 192.0.2.0-192.0.2.255
       NATed packets: 5, deNATed packets: 5, Errors: 0
   ```

2. On server S1, verify that the server is receiving the pings from H1’s NAT-translated source IP address (192.0.2.2).

   Issue the command below, and send pings again from H1.

   > **NOTE**: For this setup, another MX device is used to represent server S1 to enable monitoring of the inbound traffic.

   ```
   user@S1> monitor traffic interface xe-1/1/1 no-resolve
   verbose output suppressed, use <detail> or <extensive> for full protocol decode
   Address resolution is OFF.
   Listening on xe-1/1/1, capture size 96 bytes
   ```

```
23:28:28.577377  In IP 192.0.2.2 > 192.168.1.2: ICMP echo request, id 3293, seq 0, length 64
23:28:28.577405 Out IP 192.168.1.2 > 192.0.2.2: ICMP echo reply, id 3293, seq 0, length 64
23:28:29.579253  In IP 192.0.2.2 > 192.168.1.2: ICMP echo request, id 3293, seq 1, length 64
23:28:29.579278 Out IP 192.168.1.2 > 192.0.2.2: ICMP echo reply, id 3293, seq 1, length 64
23:28:30.579275  In IP 192.0.2.2 > 192.168.1.2: ICMP echo request, id 3293, seq 2, length 64
23:28:30.579302 Out IP 192.168.1.2 > 192.0.2.2: ICMP echo reply, id 3293, seq 2, length 64
23:28:31.580279  In IP 192.0.2.2 > 192.168.1.2: ICMP echo request, id 3293, seq 3, length 64
23:28:31.580305 Out IP 192.168.1.2 > 192.0.2.2: ICMP echo reply, id 3293, seq 3, length 64
23:28:32.581266  In IP 192.0.2.2 > 192.168.1.2: ICMP echo request, id 3293, seq 4, length 64
23:28:32.581293 Out IP 192.168.1.2 > 192.0.2.2: ICMP echo reply, id 3293, seq 4, length 64
^C
10 packets received by filter
0 packets dropped by kernel
```

Meaning

Step 1 above confirms that the inline NAT service parameters and interface-style service set are correctly implemented. Step 2 above confirms that server S1 is correctly receiving H1’s pings from its NAT-translated source IP address.

SEE ALSO

Understanding Service Sets | 8

Day One: CGNAT Up and Running on the MX Series

Example: Configuring Inline Network Address Translation—Route-Based Method

IN THIS SECTION

- Requirements | 310
- Overview and Topology | 310
- Configuration | 311
- Verification | 317

This configuration example illustrates how to configure route-based inline network address translation (NAT) on MX Series devices using `si-` (service-inline) interfaces with next-hop style service-sets.

This topic covers:

Requirements

This example uses the following hardware and software components:

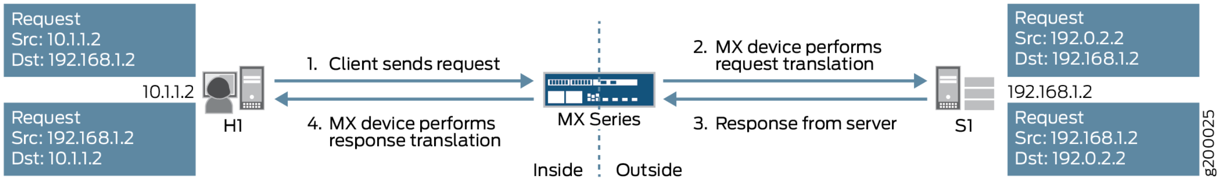
- MX Series router with a Modular Port Concentrator (MPC) line card
- Junos OS Release 11.4R1 or higher

Overview and Topology

As of Junos OS Release 11.4R1, MPC line cards can perform some services without the need of a dedicated services card, such as an MS-MPC. Inline services generally provide better performance than using a services card, however their functionality tends to be more basic. For example, inline NAT supports only static NAT.

In this example, an MX Series device with an MPC line card provides inline source NAT services to traffic flowing between two end hosts. The topology for this scenario is shown in Figure 19 on page 310

Figure 19: Inline Source NAT Using an MX Series Device with an MPC



As shown in the figure, host H1 sends traffic towards server S1. The MX Series device performs source NAT to translate H1's source IP address from 10.1.1.2 to 192.0.2.2. Server S1 then sends return traffic to host H1 using the destination IP address 192.0.2.2, and the MX Series device reverts H1's IP address back to 10.1.1.2.

The following configuration elements are used in this scenario:

- Inline service interface—a virtual interface that resides on the Packet Forwarding Engine of the MPC. To access services, traffic flows in and out of these `si-` (service-inline) interfaces.
- Service set—defines the service(s) to be performed, and identifies which inline interface(s) will feed traffic into and out of the service set. There are two ways to implement service sets:
  - Interface-style—an interface-based method, where packets arriving at an interface are forwarded through the inline service.

  - Next-hop-style—a route-based method, where static routes are used to forward packets destined for a specific destination through the inline service.

  This example uses the next-hop-style service set.

- NAT rule—uses an if-then structure (similar to firewall filters) to define matching conditions and then apply address translation to the matching traffic.

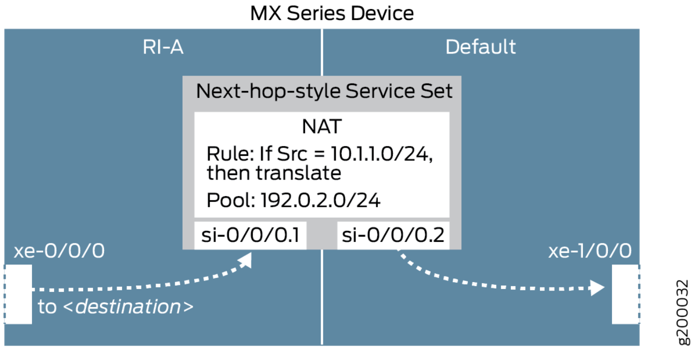
- NAT pool—a user-defined set of IP addresses that are used by the NAT rule for translation.

- Routing instance—a collection of routing tables, interfaces, and routing protocol parameters that run separate from the main (default) routing instance.

  Route-based inline NAT is typically used in scenarios that involve routing instances.

These elements come together as shown in Figure 20 on page 311 .

Figure 20: Route-Based Inline Source NAT



Configuration

IN THIS SECTION

- CLI Quick Configuration | 312
- Configure Physical Interfaces | 313
- Enable Inline Services and Create an Inline Interface | 313
- Configure Routing Instance and Identify Traffic to Send Through Inline NAT Service | 314
- Configure NAT Rule and Pool | 314
- Configure the (Next-hop-style) Service Set | 315

To configure inline NAT using a next-hop-style service set, perform these tasks:

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the [edit] hierarchy level.

```
## Configure interfaces ##
set interfaces xe-0/0/0 unit 0 family inet address 10.1.1.1/24
set interfaces xe-0/0/0 description INSIDE
set interfaces xe-1/0/0 unit 0 family inet address 192.168.1.1/24
set interfaces xe-1/0/0 description OUTSIDE
## Enable inline services, create an si- interface, reserve bandwidth ##
set chassis fpc 0 pic 0 inline-services bandwidth 1g
set interfaces si-0/0/0 unit 1 family inet
set interfaces si-0/0/0 unit 1 service-domain inside
set interfaces si-0/0/0 unit 2 family inet
set interfaces si-0/0/0 unit 2 service-domain outside
## Configure routing instance, feed traffic into the inline NAT service ##
set routing-instances RI-A instance-type virtual-router
set routing-instances RI-A interface xe-0/0/0.0
set routing-instances RI-A interface si-0/0/0.1
set routing-instances RI-A routing-options static route 192.168.1.2/32 next-hop si-0/0/0.1
## Configure a NAT rule and pool ##
set services nat rule SRC-NAT1 match-direction input
set services nat rule SRC-NAT1 term r1 from source-address 10.1.1.0/24
set services nat rule SRC-NAT1 term r1 then translated translation-type basic-nat44
set services nat rule SRC-NAT1 term r1 then translated source-pool p1
set services nat pool p1 address 192.0.2.0/24
## Configure the (next-hop-style) service set ##
set services service-set NH-STYLE-SS-NAT1 nat-rules SRC-NAT1
set services service-set NH-STYLE-SS-NAT1 next-hop-service inside-service-interface si-0/0/0.1
set services service-set NH-STYLE-SS-NAT1 next-hop-service outside-service-interface si-0/0/0.2
```

Configure Physical Interfaces

Step-by-Step Procedure

1. Configure the physical interfaces.

```
[edit interfaces]
user@MX# set xe-0/0/0 unit 0 family inet address 10.1.1.1/24
user@MX# set xe-0/0/0 description INSIDE
user@MX# set xe-1/0/0 unit 0 family inet address 192.168.1.1/24
user@MX# set xe-1/0/0 description OUTSIDE
```

Enable Inline Services and Create an Inline Interface

Step-by-Step Procedure

1. Enable inline services for the relevant FPC slot and PIC slot, and define the amount of bandwidth to dedicate for inline services.

   The FPC and PIC settings here will create and map to an `si-` interface.

```
[edit chassis fpc 0 pic 0]
user@MX# set inline-services bandwidth 1g
```

2. On the `si-` interface, create two logical units. For each unit, specify the protocol family (or families) that will need NAT services, and the ’inside’ or ’outside’ interfaces for the service domain.

> **NOTE**: The FPC and PIC settings here must match the settings defined above.

```
[edit interfaces si-0/0/0]
user@MX# set unit 1 family inet
user@MX# set unit 1 service-domain inside
user@MX# set unit 2 family inet
user@MX# set unit 2 service-domain outside
```

Configure Routing Instance and Identify Traffic to Send Through Inline NAT Service

Step-by-Step Procedure

1. Configure a routing instance that includes the 'ínside' physical and `si-` interfaces, as well as a static route that identifies traffic to forward into the inline NAT service through the `si-` interface.

   For simplicity, the static route used here simply identifies server S1.

   ```
   [edit routing-instances]
   user@MX# set RI-A instance-type virtual-router
   user@MX# set RI-A interface xe-0/0/0.0
   user@MX# set RI-A interface si-0/0/0.1
   user@MX# set RI-A routing-options static route 192.168.1.2/32 next-hop si-0/0/0.1
   ```

Configure NAT Rule and Pool

Step-by-Step Procedure

1. Configure a NAT rule that matches on traffic arriving at the MX device from H1’s subnet (10.1.1.0/24), translates it using basic IPv4 NAT, and uses an IP address from pool `p1`.

   ```
   [edit services nat]
   user@MX# set rule SRC-NAT1 match-direction input
   user@MX# set rule SRC-NAT1 term r1 from source-address 10.1.1.0/24
   user@MX# set rule SRC-NAT1 term r1 then translated translation-type basic-nat44
   user@MX# set rule SRC-NAT1 term r1 then translated source-pool p1
   ```

2. Configure the NAT pool.

   ```
   [edit services nat]
   user@MX# set pool p1 address 192.0.2.0/24
   ```

Configure the (Next-hop-style) Service Set

Step-by-Step Procedure

1. Configure a service set that uses the inline NAT service (`nat-rules`), and the inline interfaces defined above. Use the `next-hop-service` parameter to specify that this is a next-hop-style service set, and assign the `si-` interfaces as 'inside' and 'outside' based on their settings above.

   Traffic will flow into and out of the `si-` interfaces to access the inline NAT service.

   ```
   [edit services]
   user@MX# set service-set NH-STYLE-SS-NAT1 nat-rules SRC-NAT1
   user@MX# set service-set NH-STYLE-SS-NAT1 next-hop-service inside-service-interface si-0/0/0.1
   user@MX# set service-set NH-STYLE-SS-NAT1 next-hop-service outside-service-interface
   si-0/0/0.2
   ```

Results

```
chassis {
    fpc 0 {
        pic 0 {
            inline-services {
                bandwidth 1g;
            }
        }
    }
}

services {
    service-set NH-STYLE-SS-NAT1 {
        nat-rules SRC-NAT1;
        next-hop-service {
            inside-service-interface si-0/0/0.1;
            outside-service-interface si-0/0/0.2;
        }
    }
    nat {
        pool p1 {
            address 192.0.2.0/24;
        }
        rule SRC-NAT1 {
```

```
            match-direction input;
            term r1 {
                from {
                    source-address {
                        10.1.1.0/24;
                    }
                }
                then {
                    translated {
                        source-pool p1;
                        translation-type {
                            basic-nat44;
                        }
                    }
                }
            }
        }
    }
}

interfaces {
    si-0/0/0 {
       unit 1 {
            family inet;
            service-domain inside;
        }
        unit 2 {
            family inet;
            service-domain outside;
        }
    }
    xe-0/0/0 {
        description INSIDE;
        unit 0 {
            family inet {
                  address 10.1.1.1/24;
            }
        }
    }
    xe-1/0/0 {
        description OUTSIDE;
        unit 0 {
            family inet {
                address 192.168.1.1/24;
            }
        }
    }
}

routing-instances {
    RI-A {
        instance-type virtual-router;
        interface xe-0/0/0.0;
        interface si-0/0/0.1;
        routing-options {
            static {
                route 192.168.1.2/32 next-hop si-0/0/0.1;
            }
        }
    }
}
```

Verification

IN THIS SECTION

- Verifying Reachability from Host H1 to Server S1 | 317
- Verifying Address Translation | 318

Confirm that the configuration is working properly.

Verifying Reachability from Host H1 to Server S1

Purpose

Verify reachability between H1 and S1.

Action

On host H1, verify that the host can ping server S1.

```
user@H1> ping 192.168.1.2 count 5
PING 192.168.1.2 (192.168.1.2): 56 data bytes
64 bytes from 192.168.1.2: icmp_seq=0 ttl=63 time=0.926 ms
64 bytes from 192.168.1.2: icmp_seq=1 ttl=63 time=0.859 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=63 time=0.853 ms
64 bytes from 192.168.1.2: icmp_seq=3 ttl=63 time=0.825 ms
64 bytes from 192.168.1.2: icmp_seq=4 ttl=63 time=0.930 ms

--- 192.168.1.2 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.825/0.879/0.930/0.042 ms
```

Meaning

H1 can successfully reach S1.

Verifying Address Translation

Purpose

Verify that address translation is working correctly.

Action

1. On the MX device, verify that the inline NAT configuration details have been applied correctly.

   ```
   user@MX> show services inline nat pool
   Interface: si-0/0/0, Service set: NH-STYLE-SS-NAT1
     NAT pool: p1, Translation type: BASIC NAT44
       Address range: 192.0.2.0-192.0.2.255
       NATed packets: 5, deNATed packets: 5, Errors: 0, Skipped packets: 0
   ```

2. On server S1, verify that the server is receiving the pings from H1’s NAT-translated source IP address (192.0.2.2).

   Issue the command below, and send pings again from H1.

NOTE: For this setup, another MX device is used to represent server S1 to enable monitoring of the inbound traffic.

```
user@S1> monitor traffic interface xe-1/1/1 no-resolve
verbose output suppressed, use <detail> or <extensive> for full protocol decode
Address resolution is OFF.
Listening on xe-1/1/1, capture size 96 bytes

20:19:36.182690  In IP 192.0.2.2 > 192.168.1.2: ICMP echo request, id 4436, seq 0, length 64
20:19:36.182719 Out IP 192.168.1.2 > 192.0.2.2: ICMP echo reply, id 4436, seq 0, length 64
20:19:37.182918  In IP 192.0.2.2 > 192.168.1.2: ICMP echo request, id 4436, seq 1, length 64
20:19:37.182945 Out IP 192.168.1.2 > 192.0.2.2: ICMP echo reply, id 4436, seq 1, length 64
20:19:38.183914  In IP 192.0.2.2 > 192.168.1.2: ICMP echo request, id 4436, seq 2, length 64
20:19:38.183940 Out IP 192.168.1.2 > 192.0.2.2: ICMP echo reply, id 4436, seq 2, length 64
20:19:39.184872  In IP 192.0.2.2 > 192.168.1.2: ICMP echo request, id 4436, seq 3, length 64
20:19:39.184896 Out IP 192.168.1.2 > 192.0.2.2: ICMP echo reply, id 4436, seq 3, length 64
20:19:40.185882  In IP 192.0.2.2 > 192.168.1.2: ICMP echo request, id 4436, seq 4, length 64
20:19:40.185907 Out IP 192.168.1.2 > 192.0.2.2: ICMP echo reply, id 4436, seq 4, length 64
^C
10 packets received by filter
0 packets dropped by kernel
```

Meaning

Step 1 above confirms that the inline NAT service parameters and next-hop-style service set are correctly implemented. Step 2 above confirms that server S1 is correctly receiving H1’s pings from its NAT-translated source IP address.

SEE ALSO

Understanding Service Sets | 8

Day One: CGNAT Up and Running on the MX Series

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
15.1R1	Starting in Junos OS Release 15.1R1, inline NAT also supports `twice-basic-nat-44`

CHAPTER 16

Stateless Source Network Prefix Translation for IPv6

IN THIS CHAPTER

- Stateless Source Network Prefix Translation for IPv6 | 321

Stateless Source Network Prefix Translation for IPv6

IN THIS SECTION

- Stateless Source Network Prefix Translation for IPv6 Overview | 321
- Interoperation of Functionalities with Network Prefix Translation for IPv6 | 323
- Guidelines for Configuring Stateless Source Network Prefix Translation | 326
- Working of NPTv6 with Interface-Style and Next Hop-Style Service Sets | 327
- Example: Achieving Address Independence By Configuring Stateless Network Prefix Translation in IPv6 Networks by Using Interface-Style Service Sets | 328
- Example: Achieving Address Independence By Configuring Stateless Network Prefix Translation in IPv6 Networks by Using Next-Hop -Style Service Sets | 336

Stateless Source Network Prefix Translation for IPv6 Overview

IN THIS SECTION

- Benefits of Stateless Source Network Prefix Translation | 322
- NPTv6 | 322

Starting with Junos OS Release 15.1, you can configure stateless translation of source address prefixes in IPv6 networks (IPv6 to IPv6). This capability is supported on MX Series routers with MPCs where inline NAT is supported. To configure stateless network prefix translation for IPv6 packets (NPTv6), include the `translation-type nptv6` statement at the `[edit services nat rule` *`rule-name`* `term` *`term-name`* `then translated]` hierarchy level. The NPTv6 translator translates the source address prefix in such a way that the transport layer checksum of the packet does not need to be recomputed. NPTv6 defines a stateless method of IPv6 network prefix translation between internal and external networks. NPTv6 does not maintain the state for each node or each flow in the translator. You can use the `show services nat mappings nptv6 (internal | external)` command to view the NAT mappings for NPTv6 for internal and external addresses respectively. You can also use the `show services inline nat statistics` and `show services inline nat pool` commands to display information about inline NAT with NPTv6 configured.

Benefits of Stateless Source Network Prefix Translation

- For edge networks, you do not need to renumber the IPv6 addresses used inside the local network for interfaces, access lists, and system logging messages if:
  - The global prefixes used by the edge network are changed.
  - The IPv6 addresses are used inside the edge network or within other upstream networks (such as multihomed devices) when a site adds, drops, or changes upstream networks.
- IPv6 addresses used by the edge network do not need ingress filtering in upstream networks and do not need their customer-specific prefixes advertised to upstream networks.
- Connections that traverse the translation function are not disrupted by a reset or brief outage of an NPTv6 translator.

NPTv6

Network prefix translation for IPv6 (NPTv6) defines a stateless way of IPv6 address prefix translation between internal and external networks. NPTv6 does not maintain the state for each node or each flow in the translator. Maintenance of mapping state is not required for the address mapping of inbound or outbound packets. A stateless, transport-agnostic IPv6-to-IPv6 NPTv6 function offers the advantage of address-independence associated with IPv4-to-IPv4 NAT (NAPT44) and provides a 1:1 relationship between addresses in the *inside* and *outside* prefixes, thereby preserving end-to-end reachability at the network layer. In upstream networks, IPv6 addresses used by the edge network always contain a provider-allocated prefix.

NPTv6 is designed to provide address independence to the edge networks to achieve internal address stability, regardless of its upstream service provider networks. However, using provider-independent addresses without translation might be very expensive because the routing table enumerates the edge networks, instead of enumerating the transit domain that provides the service to the edge networks. This phenomenon can cause a massive and unmanageable route table. NPTv6 is a mechanism that effectively and cohesively provides address independence without advertising an internal network prefix to external networks. In contrast, the main objective of network address port translation (NAPT) for IPv4 (NAPT44) is to solve IPv4 address depletion, although it brings the same benefit of address independence. NAPT for IPv6, specifically NAPT66, is already supported in microkernel. However, similar to NAPT44, NAPT66 requires flow-state information to be preserved. NPTv6 provides a simple and streamlined technique to avoid as many of the limitations associated with NAPT66 as possible. It is defined to include a two-way, checksum-neutral, and an algorithmic translation function.

NPTv6 does not maintain state information for a node, flow, or a connection in the translator. Internal to external and external to internal packets are translated algorithmically using information present in the IPv6 header. As a result of its stateless nature, if multiple NPTv6 translators exist between the same two networks, the load can shift or be dynamically shared among them. Also, unlike NAPT44, because the mapping can be done in either direction, the translator does not interfere with the inbound connection establishment. Instead, a firewall can be used in conjunction with an NPTv6 translator. This behavior offers the network administrator more flexibility to specify security policy than that can be achieved with a traditional NAT.

Another advantage of NPTv6 is checksum-neutral translations. The translator does not need to rewrite the transport header for updating the checksum and does not perform port mapping. As a result, to deploy new transport layer protocols, you do not need to modify the translator. Because the transport layer is not modified, the algorithm does not interfere with encryption of the IP payload. Although NPTv6 compares favorably to NAPT44 or NAPT66 in several ways, it does not eliminate all of the architectural problems. Because NPTv6 modifies the IP headers of packets, it is not compatible with security mechanisms such as the IPsec authentication header. The use of separate internal and external prefixes creates complexity for Domain Name System (DNS) deployment. Also, those applications that require application layer gateways (ALGs) to work correctly through NAPT44 or NAPT66 devices might require similar ALGs to work through NPTv6 translators. Because NPTv6 does not maintain connection states, the failure of the translator does not impact the non-transmit power control (TPC) traffic through the server. TCP connections can be interrupted because of the change in the source IP address of a connection. Connections might be timed out and then reestablished in this case.

NPTv6 uses inline NAT. Inline NAT uses the capabilites of the Modular Port Concentrator (MPC) line card, eliminating the need for a MultiServices DPC (MS-DPC) for NAT. To configure inline NAT, you define your service interface as type `si-` (service-inline) interface. You must also reserve adequate bandwidth for the inline interface. This enables you to configure both interface service sets and next-hop service sets used for NAT. The `si-` interface serves as a *virtual service PIC*.

Interoperation of Functionalities with Network Prefix Translation for IPv6

IN THIS SECTION

- Address Mapping Algorithm | 324

- Internal to External Translation | 324
- External to Internal Translation | 324
- Checksum-Neutral Translation | 324
- Multihoming | 325
- Hairpinning | 325
- Load Balancing | 325
- ICMPv6 for NPTv6 | 325

This topic contains the following sections that describe the working behavior of different functionalities with stateless source IPv6 prefix translation and the various system conditions:

Address Mapping Algorithm

The NPTv6 translator filters the packets going out of the network and, if the source address of the packet matches with the source address defined in the rule (the `from` or source address in configuration), the source address is replaced with an address prefix from the pool defined for the rule. The next 16 bits after the prefix of the source address are replaced with the checksum-adjusted value to ensure that the checksum remains the same in the outgoing packet even though the source address is changed. During the definition of the configuration rule and pool for the packets going outside the network, a denat rule and pool are created for the translation of the destination address for the packets coming into the network.

Internal to External Translation

When a packet is going from the internal network to the external network, the IPv6 prefix in the source address of the packet (coming from inside node) is mapped to the external prefix. After checksum adjustment, the packet is routed toward the external network.

External to Internal Translation

When a packet is coming from external network to internal network, the IPv6 prefix in the destination address of the packet (coming from outside host) is mapped to the internal prefix. After checksum adjustment, the packet is routed to internal network.

Checksum-Neutral Translation

The NPTv6 translator translates the source address prefix in such a way that the transport layer checksum of the packet does not need to be recomputed. A checksum change caused by modifying part of the area covered by the checksum can be corrected by making an additional change to a different 16-bit field covered by the same checksum. This checksum neutral method first computes 1's complement checksum of the `internal-prefix` and the `external-prefix`.

For packets coming from the internal network, the adjustment is calculated as 1’s complement and it is computed as follows:

Adjustment = Internal prefix checksum – External prefix checksum.

The adjustment value is added to the 16-bit word of the source address after the prefix.

For packets coming from external network, the adjustment is 1’s complement and it is calculated as follows:

Adjustment = External prefix checksum – Internal prefix checksum.

The adjustment is added to the 16-bit word of the destination address after the translated prefix.

Multihoming

If there are two NPTv6 translators with different external IPv6 prefix configurations for the same internal IPv6 prefix, then these two NPTV6 translators will translate the same internal IPv6 network prefix to two different external IPv6 network prefixes, depending on the translator the packet traverses.

Hairpinning

When an internal node has knowledge of only the external (that is, the global address) of another internal node, it uses that address to send packet to that internal node. If such a packet is received by an NPTv6 translator, that packet is routed toward the internal network again after it undergoes source address and destination address translation.

Load Balancing

Load sharing is achieved when two translators have the same internal to external mapping configuration and packet load is shared between them. How the load balancing is achieved is beyond the scope of NPTv6.

The balancing could be implemented based on subnet ID portion of the IPv6 address. There can be two si- logical interfaces with the same mapping of the internal prefix to the external prefix. Packets are routed to one of the si- logical interfaces based on the subnet ID.

ICMPv6 for NPTv6

NPTv6 ICMPv6 error generation is not supported for unmapped hextet.

Guidelines for Configuring Stateless Source Network Prefix Translation

Keep the following points in mind when you configure stateless translation of source IPv6 prefixes:

This topic contains the following sections that describe the working behavior of different functionalities with stateless source IPv6 prefix translation and the various system conditions:

- Graceful Routing Engine switchover (GRES) support is the same as for NAT44.
- Unified ISSU and nonstop software upgrade (NSSU) are not supported.
- NPTv6 deployment enables direct inbound connections to internal nodes from external networks. This mechanism causes slight vulnerability because it opens the internal nodes to attacks from outside. The stateless translation of NPTv6 makes it difficult to trace external connection requests, based on connection states. This behavior enables NAT44 networks to be well-protected against external attacks. The best option to secure an NPTv6 translator is to add a firewall above the NPTv6 translator.
- A 6rd softwire concentrator interoperates with NPTv6. All other mechanisms that do not require the application layer gateway (ALG) to change the source IP address in the payload are supported. TCP, UDP, ICMP, SSH, and Telnet are supported by the NPTv6 translator. FTP and Session Initiation Protocol (SIP) that require the ALG to change the source IP address in the payload are not supported.
- The NPTv6 pools are allocated in the external data memory. The pool data structure consists of the address prefix, prefix length, and the checksum. The size of each record is of 192 bits. For every pool, a denat pool is allocated automatically. The size of the denat pool is 192 bits. There is a total allocation of 8000 64-bit entries for NAT-processed and untranslated NPTv6 pools. This allocation comes from the 64,000 entries allocated for the inline services (JNH_APP_INLINE_SVCS).
- Chaining of inline services for interoperation of 6rd with NPTv6 is not supported.
- You need to configure a source pool and specify the `from` (source) address, while configuring NPTv6.
- The external and internal prefix lengths must be greater than or equal to /16 subnet mask and less than or equal to /112 subnet mask.
- Two different internal prefixes cannot be translated to the same external prefix.
- NPTv6 cannot be applied to IPSec and Internet Key Exchange (IKE) packets. The NPTv6 translator is bypassed in this case.
- Because the translation is of one IPv6 address prefix, there is only one address in the pool. If more than one address is configured by the user, the system does not raise any error, instead only the first address prefix of the pool is chosen for translation.

- For packets going from internal network to external network, if the internal subnet is not mapped or is set to 0xFFFF, then the datagram is discarded and an ICMP destination unreachable error is generated.

- For packets going from internal network to external network, if the 16-bit word has the adjustment added to it using the 1’s complement method and is equal to 0xFFFF, then the value is written as zero.

- For packets coming from the external network to internal network, if the 16-bit word has the adjustment subtracted from it using 1’s complement method and is equal to 0xFFFF, the 16-bit word is overwritten as zero.

- For translation of prefixes /48 or shorter, the adjustment must be added or subtracted from the first 16 bits after the /48 subnet mask, the values of which are not 0xFFFF. If the prefix is /49 or longer, then the adjustment must be added or subtracted from the first 16 bits (from 64 to 123), the values of which are not 0xFFFF.

Working of NPTv6 with Interface-Style and Next Hop-Style Service Sets

The objective is to add network prefix translation for IPv6 (NPTv6) inline service that performs stateless translation of the source IPv6 address. Consider a sample topology in which NPTv6 is implemented between an internal network with the prefix of FD01:0203:0405:/48 and an external network with the prefix of 2001:0DB8:0001:/48.

The source addresses FD01:0203:0405:/48 in the packets from a single administrative domain (internal network) destined to hosts in global network (external network) will be translated to 2001:0DB8:0001:/48. Packets destined to internal network coming from external network will have their destination address as 2001:0DB8:0001:/48. This destination address will be mapped to internal network address FD01:0203:0405:/48 and will be forwarded to the internal network host. The lengths of both the subnets are assumed to be the same for this case. If they differ the shorter one would be extended to the prefix length of the longer one by suffixing zeros.

Address mapping algorithm used for NPTv6 is checksum-neutral. The translated IP headers will generate the same IPv6 pseudo-header checksum. Checksum is calculated using the standard Internet checksum algorithm. Changes that are made during translation of the IPv6 prefix are offset by the calculated changes made to the other parts of the IPv6 address. This results in transport layers that use the Internet checksum (such as TCP and UDP) calculating the same IPv6 pseudo-header checksum for both the internal and external forms of the same datagram and avoids the need to modify transport layer headers to correct the checksum value. The algorithm can map the addresses for inbound packets and outbound packets.

The NPTv6 translator works for both fragmented packets and packets with IP options enabled. The configuration changes required for NPTv6 are covered in the next sections.

The configuration of a router to handle services is through the definition of logical service interface, service sets and service set rules. These define how the service is applied to the packets.

The inline services logical interface, si-ifl, implementation available for static v4-v4 source-address inline-NAT can be reused for inline NPTv6. The configuration for the NPTv6 implemented for MS-DPC can be modified for inline NPTv6 implementation. There are two types of service set configurations—interface style and next hop style.

For the next hop-style service, a route entry is configured to steer packets to an inline service interface. There the packet would go through the service rules. If the packet matches the service rules, it is processed as per the service rules. For the interface-style service, the service set is configured directly on the media interface affecting traffic as it leaves and enters the interface. The packets are steered to the inline service interface by the service filter applied to the media interface.

Example: Achieving Address Independence By Configuring Stateless Network Prefix Translation in IPv6 Networks by Using Interface-Style Service Sets

IN THIS SECTION

- Requirements | 329
- Overview and Topology for Stateless Network Prefix Translation in IPv6 Networks Using Interface-Style Service Sets | 329
- Configuration | 329
- Verification | 334

You can configure stateless translation of source address prefixes in IPv6 networks (IPv6 to IPv6) on MX Series routers with MPCs that support inline NAT. The NPTv6 translator translates the source address prefix in such a way that the transport layer checksum of the packet does not need to be recomputed. NPTv6 defines a stateless method of IPv6 network prefix translation between internal and external networks. NPTv6 does not maintain the state for each node or each flow in the translator. You can use the `show services nat mappings nptv6 (internal | external)` command to view the NAT mappings for NPTv6 for internal and external addresses respectively. You can also use the `show services inline nat statistics` and `show services inline nat pool` commands to display information about inline NAT with NPTv6 configured.

NOTE: This functionality is supported on MX Series routers with Trio-based FPCs (MPCs).

This example describes how to configure stateless source prefix translation for IPv6 packets using interface-style service sets on MX Series routers with MPCs, and contains the following sections:

Requirements

This example uses the following hardware and software components:

- One MX Series router with an MPC.
- Junos OS Release 15.1R1 or later for MX Series routers

Overview and Topology for Stateless Network Prefix Translation in IPv6 Networks Using Interface-Style Service Sets

For the interface style service, the service set is configured directly on the media interface affecting traffic as it leaves and enters the interface. The packets are steered to the inline service interface by the service filter applied to the media interface.

When you have defined and grouped the service rules by configuring the service-set definition, you can apply services to one or more interfaces installed on the router. When you apply the service set to an interface, it automatically ensures that packets are directed to the PIC.

Consider a sample configuration scenario in which NPTv6 is configured using interface-style service sets. An inline services interface, si-0/1/0, is configured with a bandwidth reserved for 10 gigabits per second. The si-0/1/0 interface is defined with inet6 family. A NAT address pool, nptv6_pool, is configured with the address of abcd:ef12:3456::/48. A NAT rule is applied in the input direction to perform NPTv6 translation on packets that arrive from the source address of 1234:5678:9abc::/48. For packets from the source address of 1234:5678:9abc::/48 that match the NAT rule criterion, the address from the NAT address pool is allocated. A service set, ss_nptv6, is specified with the NAT rule. A gigabit Ethernet interface, ge-5/0/0, is configured and the service set is applied to this interface.

Configuration

IN THIS SECTION

- CLI Quick Configuration | 330
- Procedure | 331
- Results | 332

To configure stateless network prefix translation for IPv6 using interface-style service sets, perform these tasks:

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them in a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the [edit] hierarchy level:

Configuring Interfaces

```
set interfaces si-0/1/0 unit 0 family inet6
```

Configuring Interfaces for Traffic to Be Handled By the Service Set

```
set interfaces ge-5/0/0 unit 0 family inet6 service input service-set nptv6-service-set
set interfaces ge-5/0/0 unit 0 family inet6 service output service-set nptv6-service-set
set interfaces ge-5/0/0 unit 0 family inet6 address 1234:5678:9abc::1/64
```

Configuring Bandwidth for the Service Inline (si-) Interface

```
set chassis fpc 0 pic 1 inline-services bandwidth 10g
```

Configuring NAT Pool and Rules

```
set services nat pool ss_nptv6_pool address abcd:ef12:3456::/48
set services nat rule ss_nptv6_rule match-direction input term t0 from source-address
1234:5678:9abc::/48
set services nat rule ss_nptv6_rule match-direction input term t0 then translated source-pool
ss_nptv6_pool
set services nat rule ss_nptv6_rule match-direction input term t0 then translated translation-
type nptv6
```

Configuring the Service Set

```
set services service-set ss_nptv6 nat-rules ss_nptv6_rule
set services service-set ss_nptv6 nat-options nptv6 icmpv6-error-messages
set services service-set ss_nptv6 interface-service service-interface si-0/1/0.0
```

Procedure

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure stateless network prefix translation for IPv6 using interface-style service sets:

1. Configure an inline services (si-) interface.

```
[edit]
user@host# set interfaces si-0/1/0 unit 0 family inet6
```

2. Configure the interfaces for traffic to be handled by the service set.

```
[edit]
user@host# set interfaces ge-5/0/0 unit 0 family inet6 service input service-set nptv6-
service-set
user@host# set interfaces ge-5/0/0 unit 0 family inet6 service output service-set nptv6-
service-set
user@host# set interfaces ge-5/0/0 unit 0 family inet6 address 1234:5678:9abc::1/64
```

3. Configure the bandwidth for the service inline (si-) interface.

```
[edit]
user@host# set chassis fpc 0 pic 1 inline-services bandwidth 10g
```

4. Configure a NAT pool and rule.

```
[edit]
user@host# set services nat pool ss_nptv6_pool address abcd:ef12:3456::/48
user@host# set services nat rule ss_nptv6_rule match-direction input term t0 from source-
address 1234:5678:9abc::/48
user@host# set services nat rule ss_nptv6_rule match-direction input term t0 then translated
source-pool ss_nptv6_pool
user@host# set services nat rule ss_nptv6_rule match-direction input term t0 then translated
translation-type nptv6
```

5. Configure the service set

```
[edit]
user@host# set services service-set ss_nptv6 nat-rules ss_nptv6_rule
user@host# set services service-set ss_nptv6 nat-options nptv6 icmpv6-error-messages
user@host# set services service-set ss_nptv6 interface-service service-interface si-0/1/0.0
```

Results

From the configuration mode, confirm your configuration by entering the `show chassis`, `show interfaces`, and `show services` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show chassis
chassis {
   fpc 0 {
      pic 1 {
         inline-services {
            bandwidth 10g;
         }
      }
   }
}


user@host# show interfaces
interfaces {
   si-0/1/0 {
       unit 0 {
          family inet6;
       }
   }
   ge-5/0/0 {
       unit 0 {
          family inet6 {
             service {
                input {
                   service-set nptv6-service-set;
                }
                output {
                   service-set nptv6-service-set;
                }
             }
             address 1234:5678:9abc::1/64;
          }
       }
   }
}

user@host# show services
services {
    service-set ss_nptv6 {
       nat-rules ss_nptv6_rule;
       nat-options {
          nptv6 {
             icmpv6-error-messages;
          }
       }
       interface-service {
          service-interface si-0/1/0.0;
       }
    }
    nat {
       pool ss_nptv6_pool {
          address abcd:ef12:3456::/48;
       }
       rule ss_nptv6_rule {
          match-direction input;
          term t0 {
             from {
                source-address {
                   1234:5678:9abc::/48;
                }
             }
             then {
                translated {
                   source-pool ss_nptv6_pool;
                   translation-type {
                      nptv6;
                   }
                }
             }
          }
       }
    }
}
```

Verification

IN THIS SECTION

- Verifying the NAT Pool Mappings | 334
- Verifying the Inline NAT Pools and Statistics | 335

To confirm that the configuration is working properly, perform the following:

Verifying the NAT Pool Mappings

Purpose

Verify the existing NAT address pools and mappings for IPv6 network prefix translation.

Action

From operational mode, use the `show services nat mappings nptv6` command:

```
user@host> show services nat mappings nptv6 internal 1111:2222:3333:aaaa:bbbb::1

    Interface       Service-set     NAT-Pool           Address Mapping
    si-0/1/0        ss_nptv6         ss_nptv6_pool   1111:2222:3333:aaaa:bbbb::1 ->
aaaa:bbbb:cccc:dddd:bbbb::1
```

```
user@host> show services nat mappings nptv6 external aaaa:bbbb:cccc:dddd:bbbb::1

    Interface       Service-set     NAT-Pool              Address Mapping
    si-0/1/0        ss_nptv6            ss_nptv6_pool   1111:2222:3333:aaaa:bbbb::1 ->
aaaa:bbbb:cccc:dddd:bbbb::1
```

Meaning

The output shows the mapping between NAT addresses and ports for IPv6 stateless network prefix translation of external and internal addresses. The address and port details that are originally sent and converted using NAT are displayed.

Verifying the Inline NAT Pools and Statistics

Purpose

Verify the inline NAT pools and statistics for IPv6 network prefix translation.

Action

From operational mode, use the `show services inline nat` command:

```
user@host> show services inline nat statistics interface si-4/0/0

Service PIC Name: si-4/0/0

    Control Plane Statistics
        ICMPv4 errors packets pass through                 :0
        ICMPv4 errors packets locally generated            :0
        ICMPv6 errors packets pass through                 :0
        ICMPv6 errors packets locally generated            :0
        Dropped packets                                    :0

     Data Plane Statistics
        NATed packets                                      :0
        deNATed packets                                    :0
        Errors                                             :0

user@host> show services inline nat pool
 Interface: si-4/0/0, Service set: ss_nptv6
    NAT pool: ss_nptv6_pool1, Translation type: NPTV6
         Address range: abcd:ef12:3456::/48
         NATed packets: 0, deNATed packets: 0, Errors: 0

    NAT pool: ss_nptv6_pool2, Translation type: NPTV6
          Address range: 1111:2222:3333::/48
          NATed packets: 0, deNATed packets: 0, Errors: 0
```

```
user@host> show services inline nat pool ss_nptv6_pool1
Interface: si-4/0/0, Service set: ss_nptv6
    NAT pool: ss_nptv6_pool1, Translation type: NPTV6
          Address range: abcd:ef12:3456::/48
          NATed packets: 0, deNATed packets: 0, Errors: 0
```

Meaning

The output shows the information about inline NAT address translations, such as the number of packets that are subject to NAT processing, the packets that are not translated, and the packets with translation errors for a specified service set and an si- interface.

Example: Achieving Address Independence By Configuring Stateless Network Prefix Translation in IPv6 Networks by Using Next-Hop -Style Service Sets

IN THIS SECTION

- Requirements | 337
- Overview and Topology of Stateless Network Prefix Translation in IPv6 Networks Using Next-Hop Style Service Sets | 337
- Configuration | 337
- Verification | 344

You can configure stateless translation of source address prefixes in IPv6 networks (IPv6 to IPv6) on MX Series routers with MPCs where inline NAT is supported. The NPTv6 translator translates the source address prefix in such a way that the transport layer checksum of the packet does not need to be recomputed. NPTv6 defines a stateless method of IPv6 network prefix translation between internal and external networks. NPTv6 does not maintain per node or per flow state in the translator. You can use the show services nat mappings nptv6 (internal | external) command to view the NAT mappings for NPTv6 for internal and external addresses respectively. You can also use the show services inline nat statistics and show services inline nat pool commands to display information about inline NAT with NPTv6 configured.

NOTE: This functionality is supported on MX Series routers with Trio-based FPCs (MPCs).

This example describes how to configure stateless source prefix translation for IPv6 packets using next-hop style service sets on MX Series routers with MPCs, and contains the following sections:

Requirements

This example uses the following hardware and software components:

- One MX Series router with an MPC.
- Junos OS Release 15.1R1 or later for MX Series routers

Overview and Topology of Stateless Network Prefix Translation in IPv6 Networks Using Next-Hop Style Service Sets

A next-hop service set is a route-based method of applying a particular service. Only packets destined for a specific next hop are serviced by the creation of explicit static routes. This configuration is useful when services need to be applied to an entire virtual private network (VPN) routing and forwarding (VRF) table, or when routing decisions determine that services need to be performed.

For the next hop style service, a route entry is configured to steer packets to an inline service interface. The packet is validated through the service rules. If the packet matches the service rules, it would be processed according to the service rules.

Consider a sample configuration scenario in which NPTv6 is configured using next-hop style service sets. An inline services interface, si-0/1/0, is configured with a bandwidth reserved for 10 gigabits per second. The si-0/1/0 interface is defined with inet6 family. A NAT address pool, nptv6_pool, is configured with the address of abcd:ef12:3456::/48. A NAT rule is applied in the input direction to perform NPTv6 translation on packets that arrive from the source address of 1234:5678:9abc::/48. For packets from the source address of 1234:5678:9abc::/48 that match the NAT rule criterion, the address from the NAT address pool is allocated. The service set is configured for forwarding next-hops with the service interface of si-0/1/0.1 associated with the service set applied inside the network. with parameters for next hop service interfaces for the inside network and si-/1/0.2 associated with the service set applied outside the network. A service set, ss_nptv6, is specified with the NAT rule. The service interface domain is specified for the si- interface with the inside service-domain configured for si-0/1/0.1 and outside service domain configured for si-0/1/0.2. A routing instance, inst1, is configured with the instance type as a VRF instance. interface si-0/1/0.1 and interface ge-5/0/0 are associated with inst1. The inside and outside interface domain matches that specified with the inside-service-interface and outside-service-interface statements. A policy is configured for NAT events with the action to reject all packets.

Configuration

IN THIS SECTION

- CLI Quick Configuration | 338

- Procedure | 339
- Results | 341

To configure stateless network prefix translation for IPv6 using next-hop style service sets, perform these tasks:

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them in a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the [edit] hierarchy level:

Configuring Inline Interfaces

```
set interfaces si-0/1/0 unit 0 family inet6
set interfaces si-0/1/0 unit 1 family inet6
set interfacessi-0/1/0 unit 1 service-domain inside
set interfaces si-0/1/0 unit 2 family inet6
set interfaces si-0/1/0 unit 2 service-domain outside
set interfaces ge-5/0/0 unit 0 family inet6 address 1234:5678:9abc::1/64
```

Configuring Bandwidth for Inline Services

```
set chassis fpc 0 pic 1 inline-services bandwidth 10g
```

Configuring NAT Pool and Rule

```
set services nat pool ss_nptv6_pool address abcd:ef12:3456::/48
set services nat rule ss_nptv6_rule match-direction input term t0 from source-address
1234:5678:9abc::/48
set services nat rule ss_nptv6_rule match-direction input term t0 then translated source-pool
ss_nptv6_pool
set services nat rule ss_nptv6_rule match-direction input term t0 then translated translation-
type nptv6
```

Configuring a Service Set

```
set services service-set ss_nptv6 nat-rules ss_nptv6_rule
set services service-set ss_nptv6 nat-options nptv6 icmpv6-error-messages
set services service-set ss_nptv6 nexthop-service inside-service-interface si-0/1/0.1
set services service-set ss_nptv6 nexthop-service outside-service-interface si-0/1/0.2
```

Configuring Routing Instances

```
set routing-instances inst1 instance-type vrf
set routing-instances inst1 interface si-0/1/0.1
set routing-instances inst1 interface ge-5/0/0.0
set routing-instances inst1 route-distinguisher 1234:5678
set routing-instances inst1 vrf-import reject-all
set routing-instances inst1 vrf-export reject-all
set routing-instances inst1 routing-options rib inst1.inet6.0 static route ::0/0 next-hop
si-0/1/0.1
```

Configuring the Policy and Action Modifier

```
set policy-options policy-statement reject-all then reject
```

Procedure

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure stateless network prefix translation for IPv6 using next-hop style service sets:

1. Configure the inline interface for NAT services.

   ```
   [edit]
   user@host# set interfaces si-0/1/0 unit 0 family inet6
   user@host# set interfaces si-0/1/0 unit 1 family inet6
   user@host# set interfacessi-0/1/0 unit 1 service-domain inside
   user@host# set interfaces si-0/1/0 unit 2 family inet6
   ```

```
user@host# set interfaces si-0/1/0 unit 2 service-domain outside
user@host# set interfaces ge-5/0/0 unit 0 family inet6 address 1234:5678:9abc::1/64
```

2. Set the bandwidth for inline services.

```
[edit]
user@host# set chassis fpc 0 pic 1 inline-services bandwidth 10g
```

3. Configure the NAT pool and rule.

```
[edit]
user@host# set services nat pool ss_nptv6_pool address abcd:ef12:3456::/48
user@host# set services nat rule ss_nptv6_rule match-direction input term t0 from source-
address 1234:5678:9abc::/48
user@host# set services nat rule ss_nptv6_rule match-direction input term t0 then translated
source-pool ss_nptv6_pool
user@host# set services nat rule ss_nptv6_rule match-direction input term t0 then translated
translation-type nptv6
```

4. Configure a service set using the NAT rule associated with the NAT pool.

```
[edit]
user@host# set services service-set ss_nptv6 nat-rules ss_nptv6_rule
user@host# set services service-set ss_nptv6 nat-options nptv6 icmpv6-error-messages
user@host# set services service-set ss_nptv6 nexthop-service inside-service-interface
si-0/1/0.1
user@host# set services service-set ss_nptv6 nexthop-service outside-service-interface
si-0/1/0.2
```

5. Configure routing instances that use the si- interfaces configured.

```
[edit]
user@host# set routing-instances inst1 instance-type vrf
user@host# set routing-instances inst1 interface si-0/1/0.1
user@host# set routing-instances inst1 interface ge-5/0/0.0
user@host# set routing-instances inst1 route-distinguisher 1234:5678
user@host# set routing-instances inst1 vrf-import reject-all
user@host# set routing-instances inst1 vrf-export reject-all
```

```
user@host# set routing-instances inst1 routing-options rib inst1.inet6.0 static route ::0/0
next-hop si-0/1/0.1
```

6. Configure the policy and the action modifier for NAT packets.

```
[edit]
user@host# set policy-options policy-statement reject-all then reject
```

Results

From the configuration mode, confirm your configuration by entering the `show chassis`, `show interfaces`, `show policy-options`, `show routing-instances`, and `show services` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show chassis
chassis {
            fpc 0 {
                    pic 1 {
                            inline-services {
                                bandwidth 10g;
                            }
                    }
            }
}

user@host# show interfaces

chassis {
            fpc 0 {
                    pic 1 {
                            inline-services {
                                bandwidth 10g;
                            }
                    }
            }
}

interfaces {
  si-0/1/0 {
                unit 0 {
                        family inet6;
                }
             unit 1 {
                  family inet6;
                  service-domain inside;
                }
             unit 2 {
                  family inet6;
                  service-domain outside;
             }
             }
    ge-5/0/0 {
            unit 0 {
                 family inet6 {
                            address 1234:5678:9abc::1/64;
                              }
                    }
            }

    }



user@host# show policy-options
policy-options {
            policy-statement reject-all {
                    then reject;
            }
}


user@host# show routing-instances
routing-instances {
            inst1 {
                    instance-type vrf;
                   interface si-0/1/0.1;
                    interface ge-5/0/0.0;
                    route-distinguisher 1234:5678;
                    vrf-import reject-all;
                    vrf-export reject-all;
                    routing-options {
                            rib inst1.inet6.0 {
                                static {
                                        route ::0/0 next-hop si-0/1/0.1;
                                }
                            }
                    }
            }
}

user@host# show services
    services {
        service-set ss_nptv6 {
            nat-rules ss_nptv6_rule;
            nat-options {
                nptv6 {
                    icmpv6-error-messages;
                }
            }
            nexthop-service {
                inside-service-interface si-0/1/0.1;
                outside-service-interface si-0/1/0.2;
            }
        }
        nat {
            pool ss_nptv6_pool {
                address abcd:ef12:3456::/48;
            }
            rule ss_nptv6_rule {
                match-direction input;
                term t0 {
                    from {
                        source-address {
                            1234:5678:9abc::/48;
                        }
                                    }
                    then {
                        translated {
                            source-pool ss_nptv6_pool;
                            translation-type {
                                nptv6;
                            }
                        }
                    }
                }
            }
        }
    }
```

Verification

IN THIS SECTION

- Verifying the NAT Pool Mappings | 344
- Verifying the Inline NAT Pools and Statistics | 345

To confirm that the configuration is working properly, perform the following:

Verifying the NAT Pool Mappings

Purpose

Verify the existing NAT address pools and mappings for IPv6 network prefix translation.

Action

From operational mode, use the `show services nat mappings nptv6` command:

```
user@host> show services nat mappings nptv6 internal 1111:2222:3333:aaaa:bbbb::1

    Interface       Service-set     NAT-Pool           Address Mapping
    si-0/1/0        ss_nptv6         ss_nptv6_pool   1111:2222:3333:aaaa:bbbb::1 ->
aaaa:bbbb:cccc:dddd:bbbb::1
```

```
user@host> show services nat mappings nptv6 external aaaa:bbbb:cccc:dddd:bbbb::1

    Interface       Service-set     NAT-Pool              Address Mapping
    si-0/1/0        ss_nptv6            ss_nptv6_pool   1111:2222:3333:aaaa:bbbb::1 ->
aaaa:bbbb:cccc:dddd:bbbb::1
```

Meaning

The output shows the information about inline NAT address translations, such as the number of packets that are subject to NAT processing, the packets that are not translated, and the packets with translation errors for a specified service set and an si- interface.

Verifying the Inline NAT Pools and Statistics

Purpose

Verify the inline NAT pools and statistics for IPv6 network prefix translation.

Action

From operational mode, use the `show services inline nat` command:

```
user@host> show services inline nat statistics interface si-4/0/0

Service PIC
Name                                                                       :si-4/0/0

    Control Plane Statistics
        ICMPv4 errors packets pass through                                    :0
        ICMPv4 errors packets locally generated                              :0
        ICMPv6 errors packets pass through                                    :0
        ICMPv6 errors packets locally generated                              :0
        Dropped packets                                                                   :0

     Data Plane Statistics
        NATed packets                                                                     :0
        deNATed packets                                                                 :0

Errors                                                                                     :0

user@host> show services inline nat pool



       Interface: si-0/1/0, Service set: ss_nptv6
           NAT pool: ss_nptv6_pool1, Translation type: NPTV6
                Address range: abcd:ef12:3456::/48
                NATed packets: 0, deNATed packets: 0, Errors: 0
```

```
          NAT pool: ss_nptv6_pool2, Translation type: NPTV6
                Address range: 1111:2222:3333::/48
                NATed packets: 0, deNATed packets: 0, Errors: 0

user@host> show services inline nat pool ss_nptv6_pool1
     Interface: si-0/1/0, Service set: ss_nptv6
           NAT pool: ss_nptv6_pool1, Translation type: NPTV6
                Address range: abcd:ef12:3456::/48
                NATed packets: 0, deNATed packets: 0, Errors: 0
```

Meaning

The output shows the mapping between NAT addresses and ports for IPv6 stateless network prefix translation of external and internal addresses. The address and port details that are originally sent and converted using NAT are displayed.

CHAPTER 17

Monitoring NAT

IN THIS CHAPTER

- Monitoring NAT | 347

Monitoring NAT

IN THIS SECTION

- Configuring NAT Session Logs | 347
- Monitoring NAT Pool Usage | 349
- Using the Enterprise-Specific Utility MIB | 350

Configuring NAT Session Logs

You can configure session logs for NAT from the CLI. By default, session open and close logs are produced. However, you can request that only one type of log be produced.

To configure NAT session logs:

1. Go to the `[edit services service-set service-set-name syslog host class classname]` hierarchy level.

```
user@host# edit services service-set service-set-name syslog host class classname
```

2. Configure NAT logging using the `nat-logs` configuration statement.

```
[edit services service-set service-set-name syslog host class classname]
user@host# set nat-logs
```

3. Configure session logging using the `session-logs` statement. Open and close logs are produced by default. Specify `open` or `close` to produce only one type of log.

```
[edit services service-set service-set-name syslog host class classname]
user@host# set session-logs
```

Or

```
[edit services service-set service-set-name syslog host class classname]
user@host# set session-logs open
```

Or

```
[edit services service-set service-set-name syslog host class classname]
user@host# set session-logs close
```

4. For NAT sessions that use secured port block allocation (PBA), enter the `pba-interim-logging interval` option.

```
[edit services service-set service-set-name syslog host class classname]
user@host# top
[edit]
user@host# set interfaces interface-name service-options pba-interim-logging-interval
```

5. Configure a /32 IP address under unit 0 of the service interface that is assigned to the service set. This is the source IP address for all syslog messages generated by the service set for the NAT session logs. If you do not configure the IP address, syslog messages are not generated.

```
[edit]
user@host# set interfaces interface-name unit 0 family inet address address
```

NOTE: If you use anything other than a /32 IP address, unwanted traffic might be sent to the service interface, which can eat up valuable CPU time on the service PIC.

SEE ALSO

Configuring System Logging for Service Sets | 24
Interim Logging for Secured Port Block Allocation | 271

Monitoring NAT Pool Usage

IN THIS SECTION

- Purpose | 349
- Action | 349

Purpose

Use the `show services nat pool detail` command to find global NAT statistics related to pool usage. This command is frequently used in conjunction with the `show services stateful-firewall statistics` command.

Action

```
user@host# show services nat pool detail

Interface: ms-1/0/0, Service set: s1
  NAT pool: dest-pool, Translation type: DNAT-44
    Address range: 10.10.10.2-10.10.10.2
  NAT pool: napt-pool, Translation type: NAPT-44
    Address range: 50.50.50.1-50.50.50.254
    Port range: 1024-63487, Ports in use: 0, Out of port errors: 0, Max ports used: 0
  NAT pool: source-dynamic-pool, Translation type: DYNAMIC NAT44
    Address range: 40.40.40.1-40.40.40.254
    Out of address errors: 0, Addresses in use: 0
  NAT pool: source-static-pool, Translation type: BASIC NAT44
    Address range: 30.30.30.1-30.30.30.254
```

SEE ALSO

Configuring Pools of Addresses and Ports for Network Address Translation Overview | 94

Using the Enterprise-Specific Utility MIB

IN THIS SECTION

- Using the Enterprise-Specific Utility MIB | 350
- Populating the Enterprise-Specific Utility MIB with Information | 351
- Stopping the SLAX Script with the CLI | 358
- Clearing the Utility MIB | 358
- Recovering from an Abnormal SLAX Script Exit or a SLAX Script Exit with the CLI | 359

Using the Enterprise-Specific Utility MIB

The enterprise-specific Utility MIB enables you to add SNMP-compliant applications information to the enterprise-specific Utility MIB. The application information includes:

- NAT mappings
- Carrier-grade NAT (CGNAT) pools
- Service set CPU utilization
- Service set memory usage
- Service set summary information
- Service set packet drop information
- Service set memory zone information
- Multiservices PIC CPU and memory utilization
- Stateful firewall flow counters
- Session application connection information
- Session analysis information
- Subscriber analysis information
- Traffic Load Balancer information

You use a delivered Stylesheet Language Alternative Syntax (SLAX) script to place applications information into the enterprise-specific Utility MIB. The script is invoked based on event policies (such as reboot of the router or switchover of Routing Engines) defined in an event script. The script can also be invoked from the command line as an op script. The script only runs on the primary Routing Engine. After the script is invoked, it polls data from the specified components at regular intervals using the XML-RPC API and writes the converted data to the Utility MIB as SNMP variables. The script automatically restarts after a configured polling cycle elapses.

Populating the Enterprise-Specific Utility MIB with Information

To use a SLAX script to populate the enterprise-specific Utility MIB with information:

1. Enable the **services-oids-slax** script.

```
user@host# set system scripts op file services-oids.slax
```

2. Configure the maximum amount of memory for the data segment during the execution of the script.

```
user@host# set event-options event-script max-database 512m
```

3. Enable the script.

```
user@host# set event-options event-script file services-oids-ev-policy.slax
```

4. (Optional) Enable the **log-stats** argument to allow sys logging of stateful firewall rate statistics when the event-script is run.

   a. Display the event policies and the arguments that can be used.

```
user@host> show event-options event-scripts polices
```

```
event-options {
    policy services-oids-done {
        events system;
        attributes-match {
            system.message matches "Completed polling cycle normally. Exiting";
        }
        then {
            event-script services-oids.slax {
                arguments {
                    max-polls 30;
                    interval 120;
                }
            }
        }
    }
    policy system-started {
        events system;
        attributes-match {
            system.message matches "Starting of initial processes complete";
        }
        then {
            event-script services-oids.slax {
                arguments {
                    max-polls 30;
                    interval 120;
                }
            }
        }
    }
}
event-options {
    policy services-oids-done {
        events system;
        attributes-match {
            system.message matches "Completed polling cycle normally. Exiting";
        }
        then {
            event-script services-oids.slax {
                arguments {
                    max-polls 30;
                    interval 120;
                }
            }
        }
    }
    policy system-started {
        events system;
        attributes-match {
            system.message matches "Starting of initial processes complete";
        }
        then {
            event-script services-oids.slax {
                arguments {
                    max-polls 30;
                    interval 120;
                }
            }
        }
    }
}
```

The `log-stats` argument does not appear, so you must enable it.

b. Start the Linux shell.

```
user@host>  start shell
```

```
%
```

c. Open the **/var/db/scripts/event/services-oids-eve-policy.slax** file for editing.

```
<event-options> {
        /*
         * This policy detects when the services-oids.slax script ends, then restarts it.
         */
        <policy> {
            <name> "services-oids-done";
            <events> "system";
            <attributes-match> {
                <from-event-attribute> "system.message";
                <condition> "matches";
                <to-event-attribute-value> "Completed polling cycle normally. Exiting";
            }
            <then> {
                <event-script> {
                    <name> "services-oids.slax";
                    <arguments> {
                        <name>"max-polls";
                        <value>"30";
                    }
                    <arguments> {
                        <name>"interval";
                        <value>"120";
                    }
                    /*
                    <arguments> {
                        <name>"log-stats";
                        <value>"yes";
                    }
                    */
                }
            }
        }

        /*
         * This policy detects when the system has booted and kicks off the services-
oids.slax script.
         * This policy hooks the 'system started' event
         */
        <policy> {
            <name> "system-started";
            <events> "system";
            <attributes-match> {
                <from-event-attribute> "system.message";
                <condition> "matches";
                <to-event-attribute-value> "Starting of initial processes complete";
            }
            <then> {
                <event-script> {
                    <name> "services-oids.slax";
                    <arguments> {
                        <name>"max-polls";
                        <value>"30";
                    }
                    <arguments> {
                        <name>"interval";
                        <value>"120";
                    }
                    /*
                    <arguments> {
                        <name>"log-stats";
                        <value>"yes";
                    }
                    */
                }
            }
        }

}
```

d. Remove the comment enclosures (`/*` and `*/`) surrounding the `<arguments>` tags containing “`log-stats`”.

e. Exit the Linux shell and return to the CLI.

```
% exit
```

f. Load the changes you made to the event script file.

```
user@host>request system scripts event-scripts reload
```

The `log-stats` argument is available the next time the event script restarts.

5. Set up the script logging file **services-oids.log**.

```
user@host# set system syslog file services-oids.log any info
user@host# set system syslog file services-oids.log match cscript
```

6. Synchronize scripts between Routing Engines so that when a switchover of Routing Engine occurs, the event policy starts on the new primary.

   - To synchronize on a per-commit basis:

   ```
   user@host# commit synchronize scripts
   ```

   - To synchronize scripts every time you execute a **commit synchronize**:

   ```
   [edit system scripts]
   user@host# set synchronize
   user@host# commit synchronize
   ```

7. The script starts automatically at system boot, but you can manually start it with the CLI.

```
user@host> op services-oids arguments
```

Table 1 describes the arguments that you can use.

Table 12: Arguments for services-oids.slax Script

Argument	Description
clean	A value of **1** clears all Utility MIB OIDs. Use this only to clean OID tables.
clear-semaphore	A value of **1** resets the semaphore in the Utility MIB to recover from an abnormal script exit or from a manual script exit.
debug	Prints debug messages on console.
detail	Displays detailed output.
interval	Sets the number of seconds between poll cycles (default is 120).
invoke-debugger	Invokes script in debugger mode.
log-stats	**Yes** value enables sys logging of stateful firewall rate statistics (default is no).
max-polls	Sets the number of poll cycles before exiting the script (default is 30).
one-cycle-only	Value of **1** quits after one cycle of polling. Event policy does not restart the script. Use this option for testing only. The default is **0**.
signal-stop	A value of **1** stops the script and sets the semaphore, which causes the next iteration to exit.

Table 12: Arguments for services-oids.slax Script *(Continued)*

Argument	Description
silent	Prints trace messages on console if it is unset. Set it to zero-length string (“ ”) to unset it. Default is 1.
|	Pipes through a command.

8. Check the status of the script from the log file.

```
router> show /var/log/services-oids.log | no-more
```

```
Jun 27 19:51:47  wf-cheesypoofs cscript: services-oids.slax(v0.14):[info] Beginning polling
cycle.
Jun 27 19:51:47  wf-cheesypoofs cscript: services-oids.slax(v0.14):[info]  processing traffic
load-balance statistics
Jun 27 19:51:48  wf-cheesypoofs cscript: services-oids.slax(v0.14):[info]  processing cgnat
pool detail
Jun 27 19:51:48  wf-cheesypoofs cscript: services-oids.slax(v0.14):[info]  processing cgnat
mappings summary
Jun 27 19:51:48  wf-cheesypoofs cscript: services-oids.slax(v0.14):[info]  processing service-
sets summary
Jun 27 19:51:48  wf-cheesypoofs cscript: services-oids.slax(v0.14):[info]  processing service-
sets cpu-usage
Jun 27 19:51:48  wf-cheesypoofs cscript: services-oids.slax(v0.14):[info]  processing service-
sets mem-usage
Jun 27 19:51:49  wf-cheesypoofs cscript: services-oids.slax(v0.14):[info]  processing
stateful firewall statistics
Jun 27 19:51:49  wf-cheesypoofs cscript: services-oids.slax(v0.14):[info]  processing
stateful firewall flow-analysis
Jun 27 19:51:49  wf-cheesypoofs cscript: services-oids.slax(v0.14):[info]  processing
stateful firewall flows counts
Jun 27 19:51:49  wf-cheesypoofs cscript: services-oids.slax(v0.14):[info]  processing FW
policy connections/second
Jun 27 19:51:49  wf-cheesypoofs cscript: services-oids.slax(v0.14):[info]  processing FW/NAT
app connections
Jun 27 19:51:51  wf-cheesypoofs cscript: services-oids.slax(v0.14):[info]  processing service-
set packet-drops
```

```
Jun 27 19:51:51  wf-cheesypoofs cscript: services-oids.slax(v0.14):[info]  processing service-
set memory-usage zone
Jun 27 19:51:51  wf-cheesypoofs cscript: services-oids.slax(v0.14):[info]  processing service-
set policy throughput stats
Jun 27 19:51:52  wf-cheesypoofs cscript: services-oids.slax(v0.14):[info]  processing ms-pic
CPU amd Memory utilization stats
Jun 27 19:51:52  wf-cheesypoofs cscript: services-oids.slax(v0.14):[info] 1/30 Sleeping for
110 seconds.
```

9. Verify that you are getting Utility MIB OID updates.

```
router> show snmp mib walk jnxUtil ascii
```

```
. . .
jnxUtilCounter64Value."services10tcp-errors09CGN-SET-1" = 0
jnxUtilCounter64Value."services10tcp-errors09CGN-SET-2" = 0
jnxUtilCounter64Value."services10tcp-errors09CGN-SET-3" = 0
jnxUtilCounter64Value."services10udp-errors09CGN-SET-1" = 1119
jnxUtilCounter64Value."services10udp-errors09CGN-SET-2" = 0
. . .
```

To exclude the timestamp information, use

```
router> show snmp mib walk jnxUtil ascii | match Value
```

Stopping the SLAX Script with the CLI

To stop the SLAX script from the CLI:

- Issue the stop argument.

```
user@host> op services-oids signal-stop 1
```

Clearing the Utility MIB

To clear all the utility MIB OIDs:

- Issue the clean argument.

```
user@host> op services-oids clean 1
```

Recovering from an Abnormal SLAX Script Exit or a SLAX Script Exit with the CLI

To recover from an abnormal SLAX script exit or an SLAX script exit with the CLI:

- Issue the clear semaphore argument.

```
user@host> op services-oids clear-semaphore 1
```

RELATED DOCUMENTATION

SLAX Overview

CHAPTER 18

Packet Translation and GRE Tunneling

IN THIS CHAPTER

- Packet Translation and GRE Tunneling | 360

Packet Translation and GRE Tunneling

IN THIS SECTION

- Packet Translation and GRE Tunneling-Overview | 360
- Encapsulation Process (Edge Router-to-PaaS Server) | 361
- De-encapsulation Process (PaaS Server to Edge Router) | 365

Packet Translation and GRE Tunneling-Overview

IN THIS SECTION

- Benefits of packet translation and GRE tunneling by Enterprise Edge Routers | 361

MX routers, when deployed as Enterprise edge routers, form part of the public cloud computing platform. The enterprise edge router tunnels the IPv4 traffic received from customer VPN to the route gateway nodes. The route gateway performs IPv4-to-IPv6 translation and sends the translated packets to the Platform-as-a-service (PaaS) servers. PaaS is a complete development and deployment environment in the cloud, with resources that enable enterprises to deliver applications ranging from simple cloud-based applications to sophisticated, cloud-enabled enterprise applications.

Starting in Junos OS Release 21.2R1, as part of upgrading the customer network for PaaS services, we support enhancement to your enterprise edge routers (MX routers). You can configure your edge routers to enable translation (IPv4 to IPv6 and IPv6 to IPv4) and GRE tunneling of the translated packets through the JET APIs. The edge routers now provide access to a Private Link Service offered as Platform as a Service (PaaS), bypassing the data center gateways.

For complete information about preparing the gateway devices to interact with JET APIs, see the JET API Guide.

Benefits of packet translation and GRE tunneling by Enterprise Edge Routers

- Bypass data center gateways
- Asymmetric tunneling with respect to encapsulation and de-encapsulation
- Decoupled translation and tunnel encapsulation allows the customer to choose a different encapsulation in future with minimal software changes and also to probe forwarding path separately

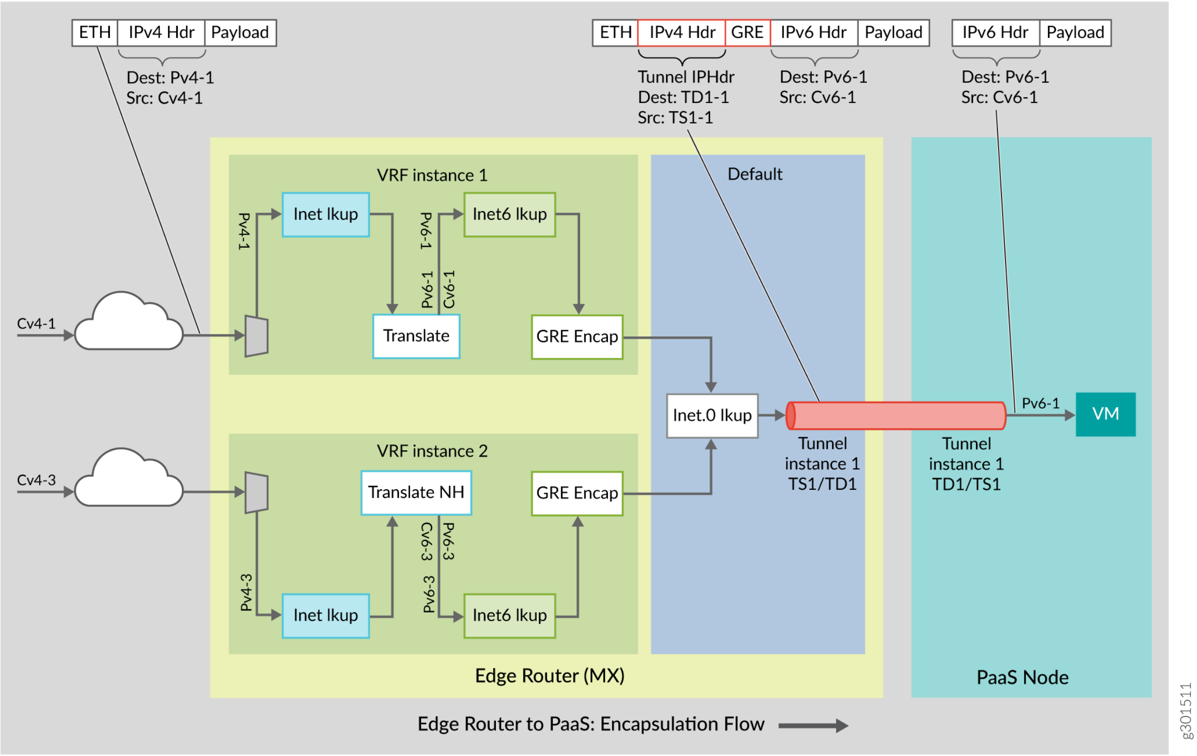
Encapsulation Process (Edge Router-to-PaaS Server)

IN THIS SECTION

- Understanding Profiles and Decoupling of Profiles | 362
- Understanding the Forwarding Path | 363
- IPv4-to-IPv6 Translation | 363
- GRE Encapsulation | 364

Figure 21 on page 362 illustrates the flow of packets from the edge router to the PaaS server.

Figure 21: Encapsulation Flow (Edge Router to PaaS Server)



IPv4 packets are translated to IPv6 (IPv4 header replaced with new IPv6 header) based on translation rule defined per destination IPv4 prefix by the controller through the PRPD API. The customer VRF inet table is looked up with IPv4 destination for the translation.

GRE Tunnel encapsulation profile is defined for the translated IPv6 packets by the controller through the PRPD API . The translated IPv6 destination is looked up in the customer VRF inet6 table for tunnel encapsulation. Multiple prefixes may use the same tunnel.

After GRE tunnel encapsulation, outer IP tunnel destination is looked up in master instance inet.0 table for nexthop L2 encapsulation. .

The process is described in detail in the following sections.

Understanding Profiles and Decoupling of Profiles

The edge router translates the packets based on the parameters defined in the translation profile. The parameters include translation type (IPv4 to IPv6 or IPv6 to IPv4), algorithm type, prefixes, and other related information.

The packets are encapsulated according to the parameters defined in the tunnel encapsulation profile.

In order to implement decoupling of the profiles, two separate routes are added by the data controller for translation and tunnel encapsulation:

- IPv4 RouteAdd() or RouteUpdate() programs the destination route in VRF inet table with the action as translate.
- IPv6 RouteAdd() or RouteUpdate() programs the translated IPv6 route in the VRF inet6 table with the action as GRE encapsulation (outer IPv4 tunnel header and GRE header)

Understanding the Forwarding Path

The edge router performs, three route lookups per IPv4 packet.

- Route lookup in the VRF inet table—The IPv4 header is translated into IPv6 header. The controller adds the IPv4 route.
- Route lookup in the VRF inet6 table—The translated IPv6 destination address is looked up to get the tunnel encapsulation profile which adds GRE and tunnel IPv4 header on top of the inner IPv6 header. The controller adds the IPv6 route.
- Route lookup in master instance—The master instance inet table is looked up for routing the tunnel packets to the tunnel destination. IGP adds the route.

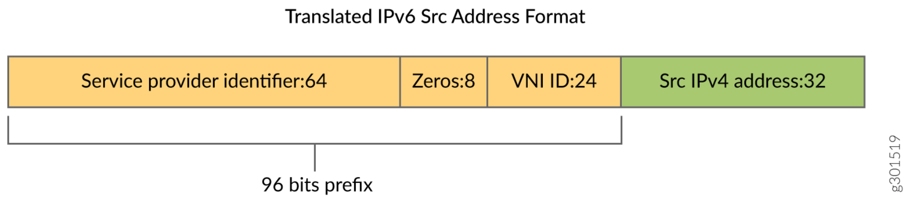
IPv4-to-IPv6 Translation

The translation details are as follows:

- If IPv4 is fragmented or has IP options, it is discarded.
- The packet type changes from IPv4 to IPv6.
- ToS / DSCP fields are copied.
- IPv6 Hop Limit is set to IPv4 TTL.
- The payload protocol is copied (without error/ inconsistency checks).
- IPv6 packet destination address is set to TranslationDestinationIPv6.
- 96 most significant bits of the IPv6 packet source address are set to TranslationSourceIPv6Prefix.
- 32 least significant bits of the IPv6 packet source address are set to original packet IPv4 address

The translated IPv6 packet is illustrated as in Figure Figure 22 on page 364

Figure 22: Translated IPv6 Source Address Format


Translated IPv6 Src Address Format
Service provider identifier:64
Zeros:8
VNI ID:24
Src IPv4 address:32
96 bits prefix
g301519

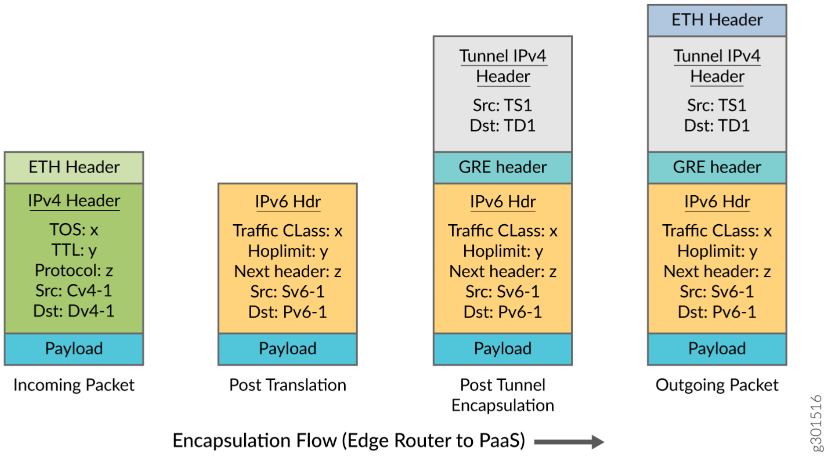
GRE Encapsulation

The controller defines the GRE tunnel encapsulation profile for the translated IPv6 destination through the PRPD API. The edge router looks up for the translated IPv6 destination in the customer VRF inet6 table for tunnel encapsulation. The details are as follows:

- Outer GRE IPv4 tunnel header is added
- GRE key is set to GREKey, which is configurable by the user
- IPv4 destination is set to TunnelDestinationIPv4
- IPv4 source is set to TunnelSourceIPv4
- **NOTE**: The number of tunnel encapsulation profiles can be less than or equal to the number of translation rules. By default, you can have a single GRE tunnel shared across multiple end customer VRFs. In PFE, there might be many GRE encapsulation routes that use the same profile parameters and thus the same tunnel nexthop. However, many translation routes may not use the same GRE route.

After encapsulation, the format of the packets at the edge router is as illustrated in Figure 23 on page 365

Figure 23: Packet Formats after Encapsulation



De-encapsulation Process (PaaS Server to Edge Router)

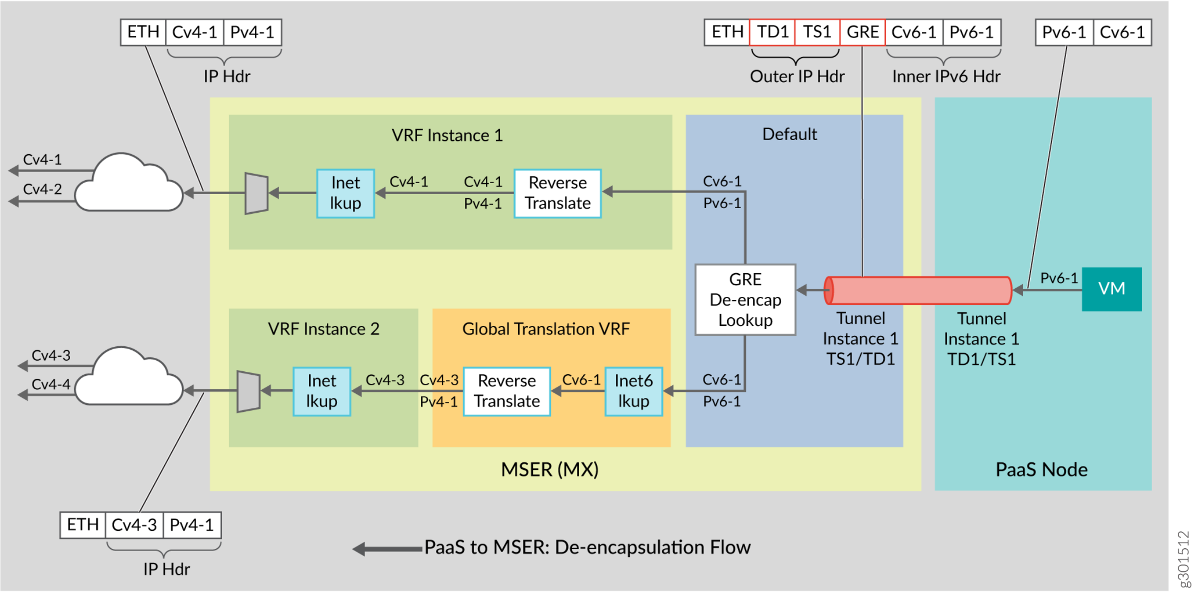
IN THIS SECTION

- Understanding Profiles and Decoupling of Profiles | 366
- Understanding the Forwarding Path | 366
- Understanding InnerIPv6Src address | 367
- GRE De-encapsulation | 368
- IPv6 to IPv4 Translation | 368

The flow of packets from the PaaS server to the edge router includes GRE de-encapsulation and reverse translation as illustrated in Figure 24 on page 366 .

The GRE encapsulated IPv6 packets received from the PaaS server, undergoes tunnel termination lookup, which de-encapsulates the GRE or IPv4 header and points to the VRF where the subsequent lookup for reverse translation happens. Based on the translation rule defined per destination IPv6 prefix by the controller, the IPv6 payload is translated (IPv6 header replaced with IPv4). After reverse translation, the translated address is looked up in the end customer VRF inet table to obtain the L2 encapsulation towards end customer network.

Figure 24: De-encapsulation Flow (PaaS server to Edge Router)



Understanding Profiles and Decoupling of Profiles

Decoupling of de-encapsulation and reverse translation profiles is essential. The controller adds two separate routes for reverse translation and tunnel de-encapsulation through separate GRPC calls.

- FlexibleTunnelAdd() programs the tunnel termination to point to the global translation VRF.
- IPv6 RouteAdd() or RouteUpdate() programs the inner destination IPv6 route in the VRF inet6 table in the global translation VRF with the action as reverse translation and targets the VRF as end customer VRF.
- The end customer VRF inet route for the translated IPv4 address is programmed by the BGP protocol.

Understanding the Forwarding Path

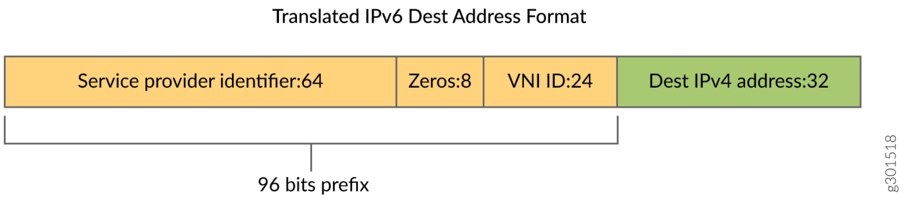
GRE encapsulated IPv6 packet received from the PaaS server undergoes tunnel termination lookup which de-encapsulates the GRE or IPv4 header and points to the VRF where the subsequent lookup for reverse translation happens.

- GRE tunnel termination lookup and tunnel de-encapsulation

  - De-encapsulation attributes form a lookup key **GRE Key, Tunnel DestinationIPv4 address, Tunnel SourceIPv4 prefix**.
  - Tunnel termination lookup identifies the customer VPN instance and removes the tunnel header (outer IPv4 header and GRE header).
  - The GRE header received from the PaaS server would have the key bit set and a 32-bit GRE key value. The forwarding lookup includes the GRE key along with the other lookup keys.
  - The packets are directed to the target VRF, which is the global translation VRF.
- Reverse translation and end customer VRF identification
  - After tunnel de-encapsulation, the traffic needs to be routed to the end customer VRF. This routing is done using the translation route lookup on the InnerDestinationIPv6 address in a Global Translation VRF, a routing instance which the customer creates for holding the IPv6 translation routes.

    You can use the InnerDestinationIPv6 address as a differentiator to identify the end customer VRF.
  - In the translation route (InnerDestinationIPv6) lookup, the reverse translation profile transforms the IPv6 header into IPv4 header and points to the end customer VRF. The lower 32 bits of the IPv6 address forms the translated IPv4 address.
- Route lookup in end customer VRF inet table
  - The translated IPv4 address is looked up in the VRF inet table for routing the packet to the customer network.

Figure 25: InnerIPv6 Destination format

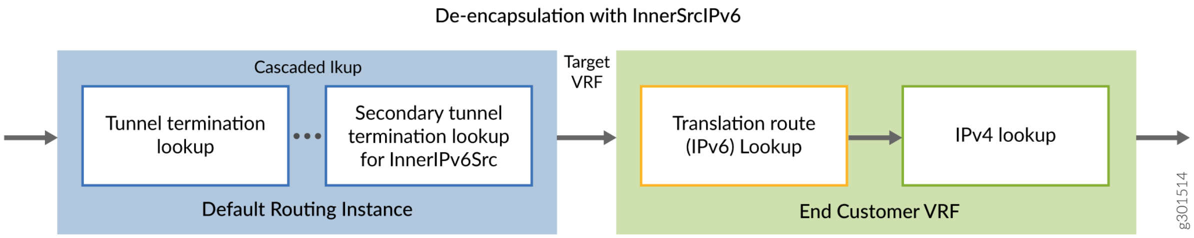

Translated IPv6 Dest Address Format
Service provider identifier:64
Zeros:8
VNI ID:24
Dest IPv4 address:32
96 bits prefix
g301518

Understanding InnerIPv6Src address

The FlexibleTunnelAdd API also supports an optional parameter, InnerSourceIPv6 field. You can use this optional parameter to include the InnerSourceIPv6 address for terminating the GRE tunnel and identifying the customer VRF.

When the InnerSourceIPv6 address is used as a differentiator, you can configure tunnel termination lookup to point to the end customer VRF by passing the end customer VRF as the target VRF in the FlexibleTunnelAdd API. In this case, the translation routes (InnerDestinationIPv6) may be programmed into the respective end customer VRF table. However, enabling the optional InnerSourceIPv6 does not restrict from using the de-encapsulation flow with global translation VRF.

Figure 26: De-encapsulation with InnerIPv6Src


De-encapsulation with InnerSrcIPv6
Cascaded lkup
Tunnel termination lookup
Secondary tunnel termination lookup for InnerIPv6Src
Default Routing Instance
Target VRF
Translation route (IPv6) Lookup
IPv4 lookup
End Customer VRF
g301514

GRE De-encapsulation

The GRE de-encapsulation process involves the following steps:

- Tunnel lookup and TargetVRF instance is determined using the lookup keys defined earlier.
- The outer GRE header is discarded.

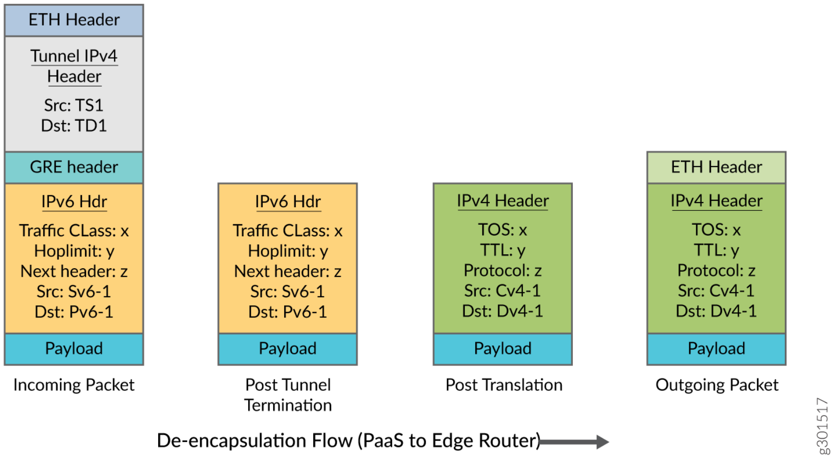
IPv6 to IPv4 Translation

The translation details are explained as follows:

- Inner packet type changes from IPv6 to IPv4. If the inner packet type is not IPv6, then the packet is dropped, and the error counter is incremented.
- Payload protocol field is copied without error or inconsistency checks.
- TC field is copied into the DSCP field
- IPv4 TTL is set to Inner IPv6 hop limit
- IPv4 destination address of the packet is derived from 32 least significant bits of the Inner IPv6 destination address
- IPv4 source address of the packet is derived from 32 least significant bits of the Inner IPv6 source address

The structure of the packets received at the edge router is as illustrated in Figure Figure 27 on page 369 .

Figure 27: Packet Formats (De-encapsulation flow)



Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
21.2R1	

3

PART

Transitioning to IPv6 Using MAP-E and MAP-T

Transitioning to IPv6 Using MAP-E and MAP-T | 371

Mapping of Address and Port with Translation (MAP-T) | 385

CHAPTER 19

Transitioning to IPv6 Using MAP-E and MAP-T

IN THIS CHAPTER

- Mapping of Address and Port with Encapsulation (MAP-E) | 371
- Equal Cost Multiple Path (ECMP) support for Mapping of Address and Port with Encapsulation (MAP-E) | 380

Mapping of Address and Port with Encapsulation (MAP-E)

IN THIS SECTION

- Understanding Mapping of Address and Port with Encapsulation (MAP-E) | 371
- Configuring Mapping of Address and Port with Encapsulation (MAP-E) | 375

Understanding Mapping of Address and Port with Encapsulation (MAP-E)

IN THIS SECTION

- Benefits of Mapping of Address and Port with Encapsulation (MAP-E) | 372
- Mapping of Address and Port with Encapsulation (MAP-E) Terminology | 372
- Mapping of Address and Port with Encapsulation (MAP-E) Functionality | 372
- Mapping of Address and Port with Encapsulation (MAP-E) Supported and Unsupported Features | 374

This topic provides an overview of Mapping of Address and Port with Encapsulation (MAP-E) feature and its benefit to service providers when used as an inline service on MX Series routers with MPC and MIC interfaces. Starting in Junos OS release 20.2R1, MAP-E softwires are supported under Next Gen

Services on either an MPC or MIC by specifying the inline services `si-1/1/0` naming convention. Starting in Junos OS release 20.3R1, MPC10E and MX2K-MPC11E support MAP-E.

Benefits of Mapping of Address and Port with Encapsulation (MAP-E)

Reduces administrative overhead and creates a scalable network infrastructure that easily supports connectivity to a large number of IPv4 subscribers over the ISP's IPv6 access network.

Mapping of Address and Port with Encapsulation (MAP-E) Terminology

1. **Border Relay (BR)**—MAP-E-enabled provider edge device in a MAP domain. A BR device has at least an IPv6-enabled interface and an IPv4 interface connected to the native IPv4 network.

2. **MAP-E Customer Edge (CE)**—MAP-E-enabled customer edge device in a MAP deployment.

3. **MAP domain**—One or more MAP-E CE devices and BR devices connected to the same virtual link.

4. **Port Set ID (PSID)**—Separate part of the transport layer port space that is denoted as port set ID.

5. **Embedded Address (EA) Bits**—EA-bits in the IPv6 address identify an IPv4 prefix or address or a shared IPv4 address and a port-set identifier.

6. **Softwire**—Tunnel between two IPv6 end-points to carry IPv4 packets or two IPv4 end-points to carry IPv6 packets.

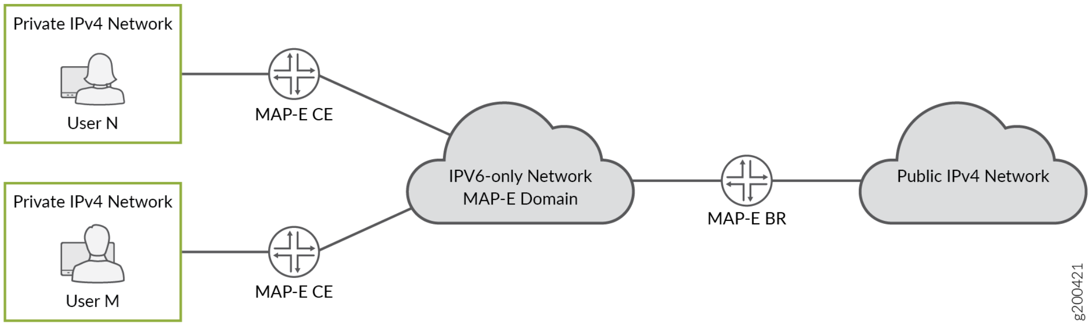
7. **Softwire Initiator (SI)**—Softwire at the customer end that encapsulates native packets and tunnels them to a softwire concentrator at the service provider.

8. **Softwire Concentrator (SC)**—Softwire that decapsulates the packets received from a softwire initiator and sends them to their destination.

Mapping of Address and Port with Encapsulation (MAP-E) Functionality

The following figure illustrates a simple MAP-E deployment scenario.

Figure 28: Sample MAP-E Deployment



In the MAP-E network topology, there are two MAP-E customer edge (CE) devices, each connected to a private IPv4 host. The MAP-E CE devices are dual stack and are capable of Network Address Port Translation (NAPT). The MAP-E CE devices connect to a MAP-E Border Relay (BR) device through an IPv6-only MAP-E network domain. The MAP-E BR device is dual stack and is connected to both a public IPv4 network and an IPv6 MAP-E network.

The MAP-E functionality is as follows:

1. The MAP-E CE devices are capable of NAPT. On receiving an IPv4 packet from the host, the MAP-E CE device performs NAT translation on the incoming IPv4 packets.

2. The NAT translated IPv4 packets are then encapsulated into IPv6 packets by the MAP-E CE device, and sent to the MAP-E BR device.

3. The IPv6 packet gets transported through the IPv6-only service provider network and reaches the MAP-E BR device.

4. On receiving the IPv6 packets, the incoming IPv6 packets are decapsulated by the MAP-E CE device and routed to the IPv4 public network.

In the reverse path, the incoming IPv4 packet is encapsulated into an IPv6 packet by the MAP-E BR device, and routed to the MAP-E CE devices.

In full reassembly of packets, the IPv4 fragments from the public IPv4 network are reassembled into a single IPv4 packet which is later encapsulated into IPv6 and routed towards MAP-E CE device. The IPv6 fragments from MAP-E CE device are reassembled into a single IPv6 packet, inner IPv4 packets are decapsulated, and forwarded to the IPv4 cloud.

Starting in Junos OS Release 22.3R1, in order to enhance the reassembly capabilities of the line cards, the line cards on MX series routers support partial reassembly of IPv4 packets for MAP-E. The MAP-E border relay device encapsulates the IPv4 packets from public IPv4 networks into IPv6 and then routes the packets to the MAP-E customer edge (CE) devices.

You must first enable IPv4 reassembly in order to configure IPv4 partial reassembly of fragments.

The following table summarises IPv4 partial reassembly capabilities.

Table 13: IPv4 Partial Reassembly Capabilities

Maximum supported fragments per flow for partial reassembly	Maximum IP packet size (in bytes) that can be partially reassembled	Maximum IP fragment size (in bytes)
64	65535	15900

When the maximum supported fragments per flow exceeds or when the maximum IP fragment size exceeds, the fragments are discarded.

Starting in Junos OS Release 22.4R1, the line cards on MX304, MX960, and MX10008 routers support full reassembly of IPv4 and IPv6 packets for Mapping of Address and Port with Encapsulation (MAP-E).

The following table summarises the enhanced IPv4 and IPv6 full reassembly capabilities.

Table 14: IPv4 and IPv6 Full Reassembly Capabilities

Maximum supported fragments per flow for full reassembly	Maximum IP packet size (in bytes) that can be fully reassembled	Maximum IP fragment size (in bytes)
16	15900	15900

When the maximum supported fragments per flow exceeds or when the maximum IP fragment size exceeds, the fragments are discarded.

Mapping of Address and Port with Encapsulation (MAP-E) Supported and Unsupported Features

Junos OS supports the following MAP-E features and functionality:

- MAP-E implementation supports line card throughput of 100 Gigabits.
- support for Inline MAP-E Border Relay (BR) solution that adheres to draft version 03 of RFC 7597

  Fully compliant with draft version 03 of RFC 7597, *Mapping of Address and Port with Encapsulation (MAP)*, when the `version-3` option is disabled at the `services softwires softwire-types map-e` *`map-e-concentrator-name`*
- Support chassis-wide scale of 250 shared MAP-E rules.
- Support the feature on all MPCs using service interfaces with 100 Gigabits.
- Ability to ping MAP-E BR IPv6 address.

- Support only next-hop style of configuration for MAP-E.
- Support reassembly of fragmented IPv4 traffic arriving from IPv4 network before encapsulating it into an IPv6 packet.
- Support fragmentation of inner IPv4 packet if the packet size after encapsulation exceeds the MAP-E maximum transmission unit (MTU).
- Packets having Internet Control Message Protocol (ICMP) payload with the following message types are accepted for MAP-E encapsulation and decapsulation:
  - Echo or Echo Reply Message of type 0 and 8
  - Timestamp or Timestamp Reply Message of type 13 and 14
  - Information Request or Information Reply Message of type 15 and 16
  - Source quench, destination_unreachable, time_exceeded, Icmp_redirect, Icmp_address_mask_reply and parameter_problem errors
- Border Relay (BR) anycast is supported.

The following features and functionality are not supported with the MAP-E feature:

- Anti-spoof check is not supported for fragmented IPv4 packets coming from a customer edge (CE) device.
- Section 8.2 of the Internet draft draft-ietf-softwire-map-03 (expires on July 28, 2013), *Mapping of Address and Port with Encapsulation (MAP)* is not supported. Instead of responding with an ICMPv6 Destination Unreachable, Source address failed ingress/egress policy (Type 1, Code 5) message, spoof packets are silently dropped and the counter is incremented.
- IPv6 reassembly is not supported.
- ICMP v6-to-v4 translation at the BR is not supported.
- Inline MAP-E with virtual routing and forwarding (VRF) is not supported.
- Inline MAP-E with inline Network Address Translation (NAT) or dual stack (DS)-Lite is not supported.
- Interface-style MAP-E configuration is not supported.

Configuring Mapping of Address and Port with Encapsulation (MAP-E)

This example shows you how to configure the MAP-E Border Relay (BR) solution using a next hop-based style of configuration.

To configure MAP-E:

1. Create service interface on the device with 100g bandwidth support.

```
[edit chassis]
user@host# set fpc 0 pic 0 inline-services bandwidth 100g
```

2. Configure the dual stack service interface unit 0.

```
[edit interfaces]
user@host# set si-0/0/0 unit 0 family inet
user@host# set si-0/0/0 unit 0 family inet6
```

3. Configure service interface inside the dual stack domain.

```
[edit interfaces]
user@host# set si-0/0/0 unit 1 family inet
user@host# set si-0/0/0 unit 1 family inet family inet6
user@host# set si-0/0/0 unit 1 service-domain inside
```

4. Configure service interface outside the dual stack domain.

```
[edit interfaces]
user@host# set si-0/0/0 unit 2 family inet
user@host# set si-0/0/0 unit 2 family inet family inet6
user@host# set si-0/0/0 unit 2 service-domain outside
```

5. Configure the IPv4-facing interface on BR.

```
[edit interfaces]
user@host# set ge-0/2/7 unit 0 family inet address 10.10.10.1/16
```

6. Configure the CPE-facing interface on BR.

```
[edit interfaces]
user@host# set ge-0/2/8 unit 0 family inet6 address 3abc::1/16
```

7. Configure the MAP-E softwire concentrator and associated parameters.

```
[edit services softwire softwire-concentrator]
user@host# set map-e swire01-rd1 version03
```

```
user@host# set map-e swire01-rd1 softwire-address 2001:db8:ffff::1
user@host# set map-e swire01-rd1 ipv4-prefix 10.10.0.0/16 mape-prefix 3040::0/16
user@host# set map-e swire01-rd1 ea-bits-len 16
user@host# set map-e swire01-rd1 psid-offset 6
user@host# set map-e swire01-rd1 psid-length 8
user@host# set map-e swire01-rd1
user@host# set mtu-ipv6 9192
user@host# set map-e swire01-rd1 v4-reassembly
```

NOTE: When configuring the MAP-E softwire concentrator, take the following into consideration:

- Possible values for `ea-bits-len` is 0 through 48.
- Possible values for `v4-prefix-len` is 0 through 32.
- If `v4-prefix-len` is 0 then `ea-bits-len` must be non-zero, and vice versa.
- It is possible that `ea-bits-len` is equal to 0, but `psid-len` is non-zero.
- If the sum of `v4-prefix-len` and `ea-bits-len` is less than 32, then the `psid-len` must be equal to the difference between 32 and the sum total of `v4-prefix-len` and `ea-bits-len`.
- The MAP-E IPv4 and IPv6 prefix must be unique per softwire concentrator.
- The MAP-E IPv4 prefix must not be 0.0.0.0
- MAP-E PSID offset has a default value of 4, and MAP-E tunnel maximum transmission unit (MTU) has a default value of 9192.

8. Configure a softwire rule to specify the direction of traffic to be tunneled and the MAP-E softwire concentrator to be used.

```
[edit services softwire]
user@host# set rule swire01-r1 match-direction input term t1 then map-e swire01-rd1
```

9. Configure the service set for MAP-E.

```
[edit services service-set]
user@host# set mape-nh-service-set softwire-rules swire01-r1
user@host# set mape-nh-service-set next-hop-service inside-service-interface si-0/0/0.1
outside-service-interface si-0/0/0.2
```

For example:

```
chassis {
    fpc 4 {
        pic 0 {
            inline-services {
                bandwidth 100g;
            }
        }
    }
    fpc 5 {
        pic 0 {
            inline-services {
                bandwidth 100g;
            }
        }
    }
}
services {
    service-set sset1 {
        softwire-rules sw-rule1;
        next-hop-service {
            inside-service-interface si-4/0/0.1;
            outside-service-interface si-4/0/0.2;
        }
    }
    service-set sset2 {
        softwire-rules sw-rule1;
        next-hop-service {
            inside-service-interface si-5/0/0.1;
            outside-service-interface si-5/0/0.2;
        }
    }
    softwire {
        softwire-concentrator {
            map-e mape-domain-1 {
                softwire-address 2001:db8:ffff::1;
                ipv4-prefix 192.0.2.0/24;
                mape-prefix 2001:db8:1234:ab00::/56;
                ea-bits-len 16;
                psid-offset 4;
                psid-length 8;
                mtu-v6 9192;
                version-03;
            }
        }
        rule sw-rule1 {
            match-direction input;
            term t1 {
                then {
                    map-e mape-domain-1;
                }
            }
        }
    }
}
interfaces {
    xe-0/1/1 {
        unit 0 {
            family inet6 {
                address 2001:db8::1/32 {
                    ndp 2001:db8:6434:0:00c0:0002:6400:3400 mac 00:11:22:33:44:55;
                }
            }
        }
    }
    xe-0/1/2 {
        unit 0 {
            family inet {
                address 100.1.1.1/24 {
                    arp 100.1.1.2 mac 00:11:22:33:44:55;
                }
            }
        }
    }
    si-4/0/0 {
        unit 1 {
            family inet;
            family inet6;
            service-domain inside;
        }
        unit 2 {
            family inet;
            family inet6;
            service-domain outside;
        }
    }
    si-5/0/0 {
        unit 1 {
            family inet6;
            service-domain inside;
        }
        unit 2 {
            family inet;
            family inet6;
            service-domain outside;
        }
    }
}
```

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
22.3R1	Starting in Junos OS Release 22.3R1, the line cards on MX series routers support partial reassembly of IPv4 fragments for MAP-E.
20.3R1	Starting in Junos OS release 20.3R1, MPC10E and MX2K-MPC11E support MAP-E.
20.2R1	Starting in Junos OS release 20.2R1, MAP-E softwires are supported under Next Gen Services on either an MPC or MIC by specifying the inline services `si-1/1/0` naming convention.

Equal Cost Multiple Path (ECMP) support for Mapping of Address and Port with Encapsulation (MAP-E)

IN THIS SECTION

- Equal Cost Multiple Path (ECMP) support for Mapping of Address and Port with Encapsulation (MAP-E) | 381

- Disabling auto-routes to support ECMP with Mapping of Address and Port with Encapsulation (MAP-E) | **381**

Equal Cost Multiple Path (ECMP) support for Mapping of Address and Port with Encapsulation (MAP-E)

IN THIS SECTION

- Benefits | **381**

This topic provides an overview of Equal Cost Multiple Path (ECMP) support for Mapping of Address and Port with Encapsulation (MAP-E) feature and its benefit to service providers when used as an inline service on MX Series routers with MPC and MIC interfaces.

In a MAP-E network topology, in the reverse path, the border relay router receives IPv4 traffic and encapsulates it in a IPv6 packet. Longer routes are used for faster matching. However, they do not facilitate EMCP load balancing on the PIC, as the routes point to a single PIC. Starting in 19.3R1, you can disable auto-routes by configuring the `disable-auto-route` statement at the `[edit services softwire softwire-concentrator map-e <domain-name>]` hierarchy, and direct the static routes to an ECMP load balancer. Hence, the packets can be distributed among different inline service interfaces.

Benefits

Enable load-balancing by distributing packets among different inline service interfaces.

Disabling auto-routes to support ECMP with Mapping of Address and Port with Encapsulation (MAP-E)

This example shows you how to disable auto-routes on a MAP-E Border Relay (BR) solution to support ECMP.

1. Create service interface on the device with 100g bandwidth support.

```
[edit chassis]
user@host# set fpc 0 pic 0 inline-services bandwidth 100g
```

2. Configure the dual stack service interface unit 0.

```
[edit interfaces]
user@host# set si-0/0/0 unit 0 family inet
user@host# set si-0/0/0 unit 0 family inet6
```

3. Configure service interface inside the dual stack domain.

```
[edit interfaces]
user@host# set si-0/0/0 unit 1 family inet
user@host# set si-0/0/0 unit 1 family inet family inet6
user@host# set si-0/0/0 unit 1 service-domain inside
```

4. Configure service interface outside the dual stack domain.

```
[edit interfaces]
user@host# set si-0/0/0 unit 2 family inet
user@host# set si-0/0/0 unit 2 family inet family inet6
user@host# set si-0/0/0 unit 2 service-domain outside
```

5. Configure the IPv4-facing interface on BR.

```
[edit interfaces]
user@host# set ge-0/2/7 unit 0 family inet address 10.10.10.1/16
```

6. Configure the CPE-facing interface on BR.

```
[edit interfaces]
user@host# set ge-0/2/8 unit 0 family inet6 address 3abc::1/16
```

7. Configure MAP-E domain 1 and associated parameters.

```
[edit services softwire softwire-concentrator]
user@host# set map-e mape-domain-1 version03
user@host# set map-e mape-domain-1 softwire-address  2001:db8:ffff::1
user@host# set map-e mape-domain-1 ipv4-prefix  192.0.2.0/24  mape-prefix 2001:db8::/32
user@host# set map-e mape-domain-1 ea-bits-len 16
user@host# set map-e mape-domain-1 psid-offset 4
user@host# set map-e mape-domain-1 psid-length 8
```

```
user@host# set map-e mape-domain-1 mtu-ipv6 9192
user@host# set map-e mape-domain-1 disable-auto-route
```

8. Configure MAP-E domain 2 and associated parameters.

```
[edit services softwire softwire-concentrator]
user@host# set map-e mape-domain-2 version03
user@host# set map-e mape-domain-2 softwire-address  2001:db8:ffff::1
user@host# set map-e mape-domain-2 ipv4-prefix  192.0.3.0/24  mape-prefix 2002:db8::/32
user@host# set map-e mape-domain-2 ea-bits-len 16
user@host# set map-e mape-domain-2 psid-offset 4
user@host# set map-e mape-domain-2 psid-length 8
user@host# set map-e mape-domain-2 mtu-ipv6 9192
user@host# set map-e mape-domain-2 disable-auto-route
```

9. Configure a softwire rule for MAP-E domain-1 to specify the direction of traffic to be tunneled.

```
[edit services softwire]
user@host# set rule sw-rule1 match-direction input term t1 then map-e mape-domain-1
```

10. Configure a softwire rule for MAP-E domain-2 to specify the direction of traffic to be tunneled.

```
[edit services softwire]
user@host# set rule sw-rule2 match-direction input term t1 then map-e mape-domain-2
```

11. Configure a single rule-set to combine both the rules.

```
[edit services softwire]
user@host# set rule-set ecmp-rules rule sw-rule1
user@host# set rule-set ecmp-rules rule sw-rule2
```

12. Configure the service set for MAP-E.

```
[edit services service-set]
user@host# set sset1 softwire-rule-sets ecmp-rules
user@host# set sset1 next-hop-service inside-service-interface si-0/0/0.1
user@host# set sset1 next-hop-service outside-service-interface si-0/0/0.2
user@host# set sset2 softwire-rule-sets ecmp-rules
```

```
user@host# set sset2 next-hop-service inside-service-interface si-0/1/0.1
user@host# set sset2 next-hop-service outside-service-interface si-0/1/0.2
```

13. Configure static routes for MAP-E BR IPv6 address.

```
[edit routing-options]
user@host# set rib inet6.0 static route 2001:db8:ffff::1/128 next-hop si-0/0/0.1
user@host# set rib inet6.0 static route 2001:db8:ffff::1/128 next-hop si-0/1/0.1
user@host# set rib inet.0 static route 192.0.2.0/24 next-hop si-0/0/0.2
user@host# set rib inet.0 static route 192.0.2.0/24 next-hop si-0/1/0.2
user@host# set rib inet.0 static route 192.0.3.0/24 next-hop si-0/0/0.2
user@host# set rib inet.0 static route 192.0.3.0/24 next-hop si-0/1/0.2
```

14. Enable load balancing.

```
[edit ]
user@host# set policy-options policy-statement LB then load-balance per-packet
user@host# set routing-options forwarding-table export LB
```

15. Verify the status of the routes.

```
[edit ]
user@host# run show route 2001:db8:ffff::1
inet6.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

2001:db8:ffff::1/128
                   *[Static/5] 00:00:12
                    >  via si-1/0/0.1
                       via si-1/1/0.1
```

The service sets of the PICs have *ecmp-rules* configured and they carry the MAP-E rules of domain-1 and domain-2. From the output, you can understand that when the `disable-auto-route` is enabled and *ecmp -rules* configured, instead of the longer auto routes, static routes are created.

RELATED DOCUMENTATION

map-e

Mapping of Address and Port with Translation (MAP-T)

IN THIS SECTION

- Understanding Mapping of Address and Port with Translation (MAP-T) | **385**
- Configuring Mapping of Address and Port using Translation (MAP-T) | **387**

Understanding Mapping of Address and Port with Translation (MAP-T)

IN THIS SECTION

- Benefits of Mapping of Address and Port with Translation (MAP-T) | **385**
- Mapping of Address and Port Using Translation (MAP-T) Terminology | **385**
- Mapping of Address and Port Using Translation (MAP-T) Functionality | **386**
- Mapping of Address and Port Using Translation (MAP-T) Supported and Unsupported Features | **387**

This topic provides an overview of Mapping of Address and Port using Translation (MAP-T) feature. The topic lists the benefit to service providers when used as an inline service on MX Series routers with MPC and MIC interfaces.

Benefits of Mapping of Address and Port with Translation (MAP-T)

The translation mode is advantageous in scenarios where the encapsulation overhead, or IPv6 operational practices rule out encapsulation (For example, use of IPv6-only servers, or reliance on IPv6 + protocol headers for traffic classification).

Mapping of Address and Port Using Translation (MAP-T) Terminology

1. **Border Relay (BR)**—MAP-T-enabled provider edge device in a MAP domain. A BR device has at least an IPv6-enabled interface and an IPv4 interface connected to the native IPv4 network.

2. **MAP-T Customer Edge(CE)**—MAP-T-enabled customer edge device in a MAP deployment.
3. **MAP domain**—One or more MAP-T CE devices and BR devices connected to the same virtual link.
4. **Port Set ID (PSID)**—Separate part of the transport layer port space that is denoted as port set ID.
5. **Embedded Address (EA) Bits**—EA-bits in the IPv6 address identify an IPv4 prefix or address or a shared IPv4 address and a port-set identifier.
6. **Softwire**—Tunnel between two IPv6 end-points to carry IPv4 packets or two IPv4 end-points to carry IPv6 packets.
7. **Softwire Initiator (SI)**—Softwire at the customer end that encapsulates native packets and tunnels them to a softwire concentrator at the service provider.
8. **Softwire Concentrator (SC)**—Softwire that decapsulates the packets received from a softwire initiator and sends them to their destination.

Mapping of Address and Port Using Translation (MAP-T) Functionality

The following figure illustrates a simple MAP-T deployment scenario.

Figure 29: Sample MAP-T Deployment

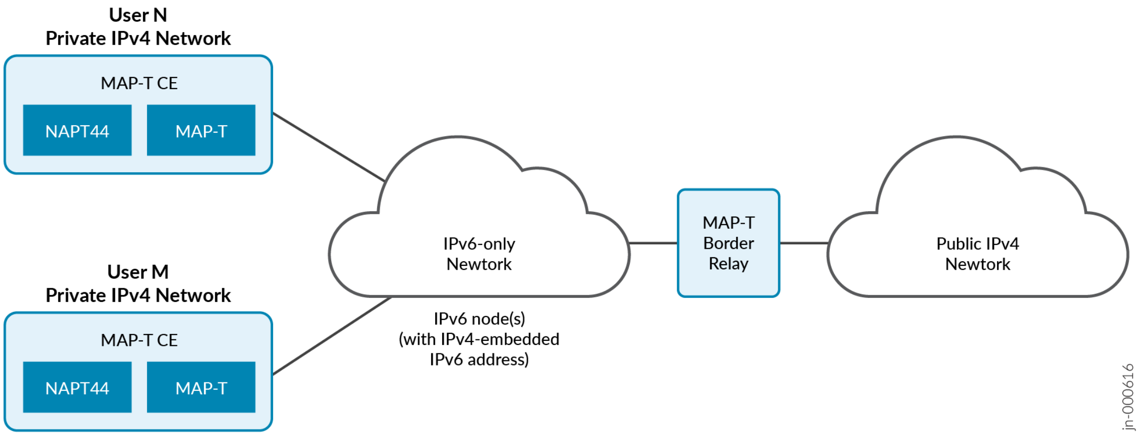


Figure 1 illustrates the MAP-T architecture. It consists of privately addressed IPv4 users (N and M) connected by means of MAP-T CEs to an IPv6 network that is equipped with one or more MAP-T BRs.

The MAP-T CE devices are dual stack and are capable of Network Address Port Translation (NAPT). The MAP-T CE devices connect to a MAP-T Border Relay (BR) device through an IPv6-only MAP-T network domain. The MAP-T BR device is dual stack and is connected to both a public IPv4 network and an IPv6 MAP-T network.

The MAP-T functionality is as follows:

1. The MAP-T CE devices are capable of NAPT. On receiving an IPv4 packet from the host, the MAP-T CE device performs NAT translation on the incoming IPv4 packets. The CE device translate the public IPv4 address and ports into an assigned IPv6 MAP source address. It sends the IPv6 packet with encoded IPv4 information toward the BR.

2. The MAP-T BR checks the source IPv6 MAP address against the configured MAP rules to ensure the correct public IPv4 address and port-range of the CE are encoded in the CE's source IPv6 MAP address. It translates the IPv6 packet into an IPv4 packet and forward it into the public domain.

In the reverse path, the incoming IPv4 packet is translated into an IPv6 packet according to MAP rules. The IPv4 destination address of the received packet is translated into an IPv6 MAP address of the CE.

Mapping of Address and Port Using Translation (MAP-T) Supported and Unsupported Features

Junos OS supports the following MAP-T features and functionality:

- TCP/UDP/ICMPv4/ICMPv6 traffic will be forwarded.

- Support only next-hop style of configuration for MAP-T.

- Support maximum packet size of 9192 bytes.

The following features and functionality are not supported with the MAP-T feature:

- IPv4/IPv6 reassembly is not supported.

- IPv6 fragmentation is not supported.

- Implementation of draft (non-standard) versions of RFC 7599 is not supported.

- Interface-style MAP-T configuration is not supported.

- MAP-T and non-MAP-T service in the same service-set is not supported.

Configuring Mapping of Address and Port using Translation (MAP-T)

This example shows you how to configure the MAP-T Border Relay (BR) solution using a next hop-based style of configuration.

To configure MAP-T:

1. Create service interface on the device with 100g bandwidth support.

   ```
   [edit chassis]
   user@host# set fpc 0 pic 0 inline-services bandwidth 100g
   ```

2. Configure the dual stack service interface unit 0.

   ```
   [edit interfaces]
   user@host# set si-0/0/0 unit 0 family inet
   user@host# set si-0/0/0 unit 0 family inet6
   ```

3. Configure service interface inside the dual stack domain.

   ```
   [edit interfaces]
   user@host# set si-0/0/0 unit 1 family inet
   user@host# set si-0/0/0 unit 1 family inet family inet6
   user@host# set si-0/0/0 unit 1 service-domain inside
   ```

4. Configure service interface outside the dual stack domain.

   ```
   [edit interfaces]
   user@host# set si-0/0/0 unit 2 family inet
   user@host# set si-0/0/0 unit 2 family inet family inet6
   user@host# set si-0/0/0 unit 2 service-domain outside
   ```

5. Configure the IPv4-facing interface on BR.

   ```
   [edit interfaces]
   user@host# set ge-0/2/7 unit 0 family inet address 10.10.10.1/16
   ```

6. Configure the CPE-facing interface on BR.

   ```
   [edit interfaces]
   user@host# set ge-0/2/8 unit 0 family inet6 address 3abc::1/16
   ```

7. Configure the MAP-T softwire concentrator and associated parameters.

   ```
   [edit services softwire softwire-concentrator]
   user@host# set map-t mapt-domain1 dmr-prefix  2001:db8:ffff::/64
   user@host# set map-t mapt-domain1 ipv4-prefix 192.0.2.0/24
   user@host# set map-t mapt-domain1 mapt-prefix 2001:db8::/32
   user@host# set map-t mapt-domain1 ea-bits-len 16
   user@host# set map-t mapt-domain1 psid-offset 6
   user@host# set map-t mapt-domain1 psid-length 8
   user@host# set map-t mapt-domain1 mtu-ipv6 9192
   ```

   NOTE:

   - The MAP-T IPv4 prefix must not be 0.0.0.0

8. Configure a softwire rule to specify the direction of traffic to be tunneled and the map-t softwire concentrator to be used.

   ```
   [edit services softwire]
   user@host# set rule sw-r1 match-direction input term t1 then map-t mapt-domain1
   ```

9. Configure the service set for map-t.

   ```
   [edit services service-set]
   user@host# set mape-nh-service-set softwire-rules swire01-r1
   user@host# set mape-nh-service-set next-hop-service inside-service-interface si-0/0/0.1
   outside-service-interface si-0/0/0.2
   ```

For example:

```
chassis {
    fpc 4 {
        pic 0 {
            inline-services {
                bandwidth 100g;
            }
        }
    }
    fpc 5 {
```

```
        pic 0 {
            inline-services {
                bandwidth 100g;
            }
        }
    }
}
services {
    service-set sset1 {
        softwire-rules sw-rule1;
        next-hop-service {
            inside-service-interface si-4/0/0.1;
            outside-service-interface si-4/0/0.2;
        }
    }
    service-set sset2 {
        softwire-rules sw-rule1;
        next-hop-service {
            inside-service-interface si-5/0/0.1;
            outside-service-interface si-5/0/0.2;
        }
    }
    softwire {
        softwire-concentrator {
            map-t mapt-domain-1 {
                softwire-address 2001:db8:ffff::1;
                ipv4-prefix 192.0.2.0/24;
                mape-prefix 2001:db8:1234:ab00::/56;
                ea-bits-len 16;
                psid-offset 4;
                psid-length 8;
                mtu-v6 9192;
                version-03;
            }
        }
        rule sw-rule1 {
            match-direction input;
            term t1 {
                then {
                    map-t mapt-domain-1;
                }
            }
        }
    }
}
interfaces {
    xe-0/1/1 {
        unit 0 {
            family inet6 {
                address 2001:db8::1/32 {
                    ndp 2001:db8:6434:0:00c0:0002:6400:3400 mac 00:11:22:33:44:55;
                }
            }
        }
    }
    xe-0/1/2 {
        unit 0 {
            family inet {
                address 100.1.1.1/24 {
                    arp 100.1.1.2 mac 00:11:22:33:44:55;
                }
            }
        }
    }
    si-4/0/0 {
        unit 1 {
            family inet;
            family inet6;
            service-domain inside;
        }
        unit 2 {
            family inet;
            family inet6;
            service-domain outside;
        }
    }
    si-5/0/0 {
        unit 1 {
            family inet6;
            service-domain inside;
        }
        unit 2 {
            family inet;
            family inet6;
            service-domain outside;
        }
    }
}
```

SEE ALSO

> No Link Title

4

PART

Transition to IPv6 With Softwires

Transition to IPv6 With 6to4 Softwires | 394

Transition to IPv6 With DS-Lite Softwires | 407

Transition to IPv6 With 6rd Softwires | 432

Transition to IPv6 With Inline Softwires | 451

Monitoring and Troubleshooting Softwires | 469

CHAPTER 20

Transition to IPv6 With 6to4 Softwires

IN THIS CHAPTER

- Softwires Configuration Overview | 394
- 6to4 Softwires | 402

Softwires Configuration Overview

IN THIS SECTION

- Tunneling Services for IPv4-to-IPv6 Transition Overview | 394
- Configuring Softwire Rules | 399
- Configuring Service Sets for Softwire | 400

Tunneling Services for IPv4-to-IPv6 Transition Overview

IN THIS SECTION

- 6to4 Overview | 395
- DS-Lite Softwires—IPv4 over IPv6 | 397
- 6rd Softwires—IPv6 over IPv4 | 398

Junos OS enables service providers to transition to IPv6 by using softwire encapsulation and decapsulation techniques. A softwire is a tunnel that is created between softwire customer premises equipment (CPE). A softwire CPE can share a unique common internal state for multiple softwires, making it a very light and scalable solution. When you use softwires, you need not maintain an interface infrastructure for each softwire, unlike a typical mesh of generic routing encapsulation (GRE) tunnels that requires you to do so. A softwire initiator at the customer end encapsulates native packets and tunnels them to a softwire concentrator at the service provider. The softwire concentrator decapsulates the packets and sends them to their destination. A softwire is created when a softwire concentrator receives the first tunneled packet of a flow and prepares the packet for flow processing. The softwire exists as long as the softwire concentrator is providing flows for routing. A flow counter is maintained; when the number of active flows is 0, the softwire is deleted. Statistics are kept for both flows and softwires.

Softwire addresses are not specifically configured under any physical or virtual interface. The number of established softwires does not affect throughput, and scalability is independent of the number of interfaces. Scalability is only limited to the number of flows that the services DPC or PIC can support.

This topic contains the following sections:

6to4 Overview

Basic 6to4

6to4 is an Internet transition mechanism for migrating from IPv4 to IPv6, a system that enables IPv6 packets to be transmitted over an IPv4 network (generally the IPv4 Internet) without the need to configure explicit tunnels. 6to4 is described in RFC 3056, *Connection of IPv6 Domains via IPv4 Clouds*. 6to4 is especially relevant during the initial phases of deployment to full, native IPv6 connectivity, because IPv6 is not required on nodes between the host and the destination. 6to4 is intended only as a transition mechanism and is not meant to be used permanently.

6to4 is supported on Multiservices 100, 400, and 500 PICs on M Series routers and on MX Series routers equipped with Multiservices DPCs and on MX240, MX480, and MX960 routers with the MX-SPC3 services card. 6rd is not supported on MX Series routers with MS-MPCs or MS-MICs.

6to4 can be used by an individual host or by a local IPv6 network. When used by a host, 6to4 must have a global IPv4 address connected, and the host is responsible for the encapsulation of outgoing IPv6 packets and the decapsulation of incoming 6to4 packets. If the host is configured to forward packets for other clients, often a local network, it is then a router.

There are two kinds of 6to4 virtual routers: border routers and relay routers.

- A 6to4 border router is an IPv6 router supporting a 6to4 pseudointerface, and is normally the border router between an IPv6 site and a wide-area IPv4 network.
- A relay router is a 6to4 router configured to support transit routing between 6to4 addresses and pure native IPv6 addresses.

In order for a 6to4 host to communicate with the native IPv6 Internet, the host’s IPv6 default gateway must be set to a 6to4 address that contains the IPv4 address of a 6to4 relay router. To avoid the need for users to set this up manually, the Anycast address of 192.88.99.1 has been allocated to send packets to a 6to4 relay router. When processed by 6to4 with the subnet and hosts fields set to zero, this IPv4 address (192.88.99.1) becomes the IPv6 address 2002:c058:6301::. To ensure BGP routing propagation, a short prefix of 192.88.99.0/24 has been allocated for routes pointed at 6to4 relay routers that use this Anycast IP address. We recommend that providers who want to provide 6to4 service to their clients or peers advertise the Anycast prefix like any other IP prefix, and route the prefix to the provider's 6to4 relay.

Packets from the IPv6 Internet to 6to4 systems must be sent to a 6to4 relay router by normal IPv6 routing methods. The specification states that such relay routers must only advertise 2002::/16 and not subdivisions of it to prevent the embedding of IPv4 routes into the routing tables of IPv6 routers. From the 6to4 relay router the packets can then be sent over the IPv4 Internet to the destination.

6to4 Anycast

Router 6to4 requires that 6to4 routers and relays are managed and configured cooperatively. In particular, 6to4 sites must configure a relay router to carry the outbound traffic, which becomes the default IPv6 router (except for 2002::/16). The objective of the Anycast variant, defined in RFC 3068, *An Anycast Prefix for 6to4 Relay Routers*, is to avoid the need for such configuration. Removing this configuration makes the solution available for small or domestic users, even those with a single host or simple home gateway instead of a border router. Removing this configuration is achieved by defining 192.88.99.1 as the default IPv4 address for a 6to4 relay, and 2002:c058:6301:: as the default IPv6 router prefix (*well-known prefix*) for a 6to4 site.

RFC 6343, *Advisory Guidelines for 6to4 Deployment*, published in August 2011, identifies a wide range of problems associated with the use of unmanaged 6to4 Anycast relay routers.

6to4 Provider-Managed Tunnels

A solution to many problems associated with unmanaged Anycast 6to4 is presented in IETF informational draft draft-kuarsingh-v6ops-6to4-provider-managed-tunnel-02, *6to 4 Provider-Managed Tunnels (PMT)*. That document, a work in progress, proposes a solution that providers can implement to exercise greater control over the routing of 6to4 traffic.

Anycast 6to4 implies a default configuration for the user site. It does not require any particular user action. It does require an IPv4 Anycast route to be in place to a relay at 192.88.99.1. Traffic does not necessarily return to the same 6to4 gateway because of the *well-known* 6to4 prefix used and advertised by all 6to4 traffic.

6to4 provider-managed tunnels (PMTs) facilitate the management of 6to4 tunnels using an Anycast configuration. 6to4 PMT enables service providers to improve 6to4 operation when network conditions provide suboptimal performance or break normal 6to4 operation. 6to4 PMT provides a stable provider prefix and forwarding environment by utilizing existing 6to4 relays with an added function of IPv6 prefix translation that controls the flow of return traffic.

The 6to4 managed tunnel model behaves like a standard 6to4 service between the customer IPv6 host or gateway and the 6to4 PMT relay (within the provider domain). The 6to4-PMT relay shares properties with 6rd (RFC5969) by decapsulating and forwarding embedded IPv6 flows, within an IPv4 packet, to the IPv6 Internet. The model provides an additional function that translates the source 6to4 prefix to a provider assigned prefix that is not found in 6rd (RFC5969) or traditional 6to4 operation. The 6to4-PMT relay provides a stateless (or stateful) mapping of the 6to4 prefix to a provider-supplied prefix by mapping the embedded IPv4 address in the 6to4 prefix to the provider prefix.

DS-Lite Softwires—IPv4 over IPv6

When an ISP begins to allocate new subscriber home IPv6 addresses and IPv6-capable equipment, dual-stack lite (DS-Lite) provides a method for the private IPv4 addresses behind the IPv6 customer edge WAN equipment to reach the IPv4 network. DS-Lite enables IPv4 customers to continue to access the Internet using their current hardware by using a softwire initiator, referred to as a Basic Bridging Broadband (B4), at the customer edge to encapsulate IPv4 packets into IPv6 packets and tunnel them over an IPv6 network to a softwire concentrator, referred to as an Address Family Transition Router (AFTR), for decapsulation. DS-Lite creates the IPv6 softwires that terminate on the services PIC. Packets coming out of the softwire can then have other services such as NAT applied on them.

DS-Lite is supported on MS-DPCs and MS-100, MS-400, and MS-500 MultiServices PICS. Starting in Junos OS release 20.2R1, DS-Lite is supported Next Gen Services on MX240, Mx480 and MX960 routers with the MX-SPC3.Starting in Junos OS release 17.4R1, DS-Lite is supported on MX Series routers with MS-MPCs and MS-MICs. Starting in Junos OS release 18.2R1, DS-lite is supported on AMS interfaces.Starting in Junos OS release 19.2R1, DS-Lite is supported on MX Virtual Chassis and MX Broadband Network Gateway (BNG) routers.

NOTE: IPv6 Provider Edge , or MPLS-enabled IPv6, is available for ISPs with MPLS-enabled networks. These networks now can use Multiprotocol BGP (MP-BGP) to provide connectivity between the DS-Lite B4 and AFTR (or any two IPv6 nodes). DS-Lite properly handles encapsulation and decapsulation despite the presence of additional MPLS header information.

For more information on DS-Lite softwires, see the IETF draft *Dual Stack Lite Broadband Deployments Following IPv4 Exhaustion*.

NOTE: The most recent IETF draft documentation for DS-Lite uses new terminology:

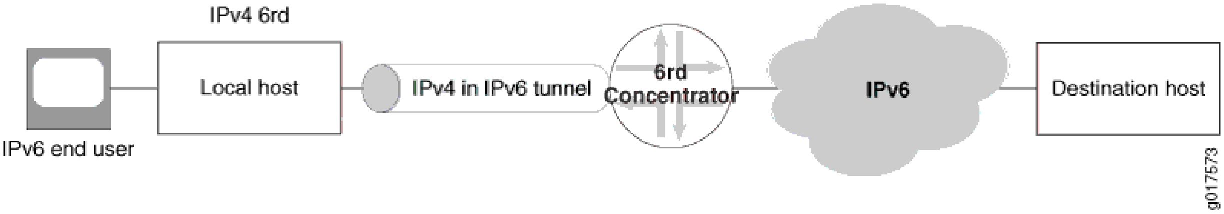
- The term *softwire initiator* has been replaced by *B4*.
- The term *softwire concentrator* has been replaced by *AFTR*.

The Junos OS documentation generally uses the original terms when discussing configuration in order to be consistent with the command-line interface (CLI) statements used to configure DS-Lite.

6rd Softwires—IPv6 over IPv4

6rd softwire flow is shown in Figure 30 on page 398 .

Figure 30: 6rd Softwire Flow



Junos OS supports a 6rd softwire concentrator on a services DPC or PIC to facilitate rapid deployment of IPv6 service to subscribers on native IPv4 customer edge WANs. IPv6 packets are encapsulated in IPv4 packets by a softwire initiator at the customer edge WAN. These packets are tunneled to a softwire concentrator residing on an MS-DPC or MX-SPC3 (branch relay). A softwire is created when IPv4 packets containing IPv6 destination information are received at the softwire concentrator, which decapsulates IPv6 packets and forwards them for IPv6 routing. All of these functions are performed in a single pass of the services PIC.

In the reverse path, IPv6 packets are sent to the services DPC where they are encapsulated in IPv4 packets corresponding to the proper softwire and sent to the customer edge WAN.

The softwire concentrator creates softwires as the IPv4 packets are received from the customer edge WAN side or IPv6 packets are received from the Internet. A 6rd softwire on the Services DPC is identified by the 3-tuple containing the service set ID, customer edge softwire initiator IPv4 address, and softwire concentrator IPv4 address. IPv6 flows are also created for the encapsulated IPv6 payload, and are associated with the specific softwire that carried them in the first place. When the last IPv6 flow associated with a softwire ends, the softwire is deleted. This simplifies configuration and there is no need to create or manage tunnel interfaces.

6rd is supported on Multiservices 100, 400, and 500 PICs on M Series routers, and on MX Series routers equipped with Multiservices DPCs. 6rd is not supported on MX Series routers with MS-MPCs or MS-MICs.

Junos OS supports inline 6rd and 6to4 on Modular Port Concentrator (MPC) line cards.

For more information on 6rd softwires, see RFC 5969, *IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) -- Protocol Specification*.

SEE ALSO

Junos Address Aware Network Addressing Overview | **53**

Configuring a 6rd Softwire Concentrator | **432**

Configuring Inline 6rd | **452**

Configuring Softwire Rules

You configure softwire rules to instruct the router how to direct traffic to the addresses specified for 6rd or DS-Lite softwire concentrators. Softwire rules do not perform any filtration of the traffic. They do not include a `from` statement, and the only option in the `then` statement is to specify the address of the 6rd or DS-Lite softwire concentrator.

Softwire rules are supported on Multiservices 100, 400, and 500 PICs on M Series routers, and on MX Series routers equipped with Multiservices DPCs. Starting in Junos OS release 17.4R1, softwire rules for DS-Lite are supported on MX Series routers with MS-MPCs and MS-MICs.Starting in Junos OS release 19.2R1, softwire rules for DS-Lite are supported on MX Virtual Chassis and MX Broadband Network Gateway (BNG) routers.

You can create a softwire rule consisting of one or more terms and associate a particular 6rd or DS-Lite softwire concentrator with each term. You can include the softwire rule in service sets along with other services rules.

To configure a softwire rule:

1. Assign a name to the rule.

```
[edit services softwire ]
user@host# edit rule rule-name
```

2. Specify the match direction.

```
[edit services softwire rule rule-name]
user@host# set match-direction (input | output)
```

3. Assign a name for the first term.

```
[edit services softwire rule rule-name]
user@host# edit term term-name
```

4. Associate a 6rd or DS-Lite softwire concentrator with this term.

```
[edit services softwire rule rule-name term term-name]
user@host# set then ds-lite name
```

Or

```
user@host# set then v6rd v6rd-softwire-concentator
```

5. Repeat Steps 3 and 4 for as many additional terms as needed.

SEE ALSO

Configuring a 6rd Softwire Concentrator | 432

Configuring Service Sets for Softwire

You must include softwire rules or a softwire rule set in a service set to enable softwire processing. You must include a stateful firewall rule for DS-Lite.

Softwire rules are supported on Multiservices 100, 400, and 500 PICs on M Series routers, and on MX Series routers equipped with Multiservices DPCs. Starting in Junos OS release 17.4R1, softwire rules for DS-Lite are supported on MX Series routers with MS-MPCs and MS-MICs.Starting in Junos OS release 19.2R1, softwire rules for DS-Lite are supported on MX Virtual Chassis and MX Broadband Network Gateway (BNG) routers.

To configure service sets for softwire:

1. Include a softwire rule or rule set in the service set.

```
[edit services service-set service-set-name]
user@host# set softwire-rules rule softwire-rule-name
```

2. When using a 6rd softwire, include a stateful-firewall rule.

```
[edit services service-set service-set-name]
user@host# set stateful-firewall-rules softwire-rule-name
```

3. You can include a NAT rule for flows originated by DS-Lite softwires.

> **NOTE**:
>
> Currently a NAT rule configuration is required with a DS-Lite softwire configuration when you use interface service set configurations; NAT is not required when using next-hop service set configurations. NAT processing from IPv4 to IPv6 address pools and vice versa is not currently supported. FTP, HTTP, and RSTP are supported.

> **NOTE**: With a DS-Lite softwire concentrator, if you configure stateful firewall rules without configuring NAT rules, using an interface service set causes the ICMP echo reply messages to be not sent correctly to DS-Lite. This behavior occurs if you apply a service set to both inet and inet6 families. In such a scenario, the traffic that is not destined to the DS-Lite softwire concentrator is also processed by the service set and the packets might be dropped, although the service set must not process such packets.
>
> To prevent the problem to incorrect processing of traffic applicable for DS-Lite, you must configure a next-hop style service set and not an interface style service set. This problem does not occur when you configure NAT rules with interface service sets for DS-Lite.

For further information, see “"Configuring Service Rules" on page 27 .”

SEE ALSO

Example: Configuring DS-Lite and 6rd in the Same Service Set | 418

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
20.2R1	Starting in Junos OS release 20.2R1, DS-Lite is supported Next Gen Services on MX240, Mx480 and MX960 routers with the MX-SPC3.

19.2R1	Starting in Junos OS release 19.2R1, DS-Lite is supported on MX Virtual Chassis and MX Broadband Network Gateway (BNG) routers.
19.2R1	Starting in Junos OS release 19.2R1, softwire rules for DS-Lite are supported on MX Virtual Chassis and MX Broadband Network Gateway (BNG) routers.
19.2R1	Starting in Junos OS release 19.2R1, softwire rules for DS-Lite are supported on MX Virtual Chassis and MX Broadband Network Gateway (BNG) routers.
18.2R1	Starting in Junos OS release 18.2R1, DS-lite is supported on AMS interfaces.
17.4R1	Starting in Junos OS release 17.4R1, DS-Lite is supported on MX Series routers with MS-MPCs and MS-MICs.
17.4R1	Starting in Junos OS release 17.4R1, softwire rules for DS-Lite are supported on MX Series routers with MS-MPCs and MS-MICs.
17.4R1	Starting in Junos OS release 17.4R1, softwire rules for DS-Lite are supported on MX Series routers with MS-MPCs and MS-MICs.

6to4 Softwires

IN THIS SECTION

- Configuring a 6to4 Provider-Managed Tunnel | 402

Configuring a 6to4 Provider-Managed Tunnel

When configuring a 6to4 provider-managed tunnel (PMT), replace the Anycast destination with the address of a managed relay in the provider network.

6to4 tunnels are supported on Multiservices 100, 400, and 500 PICs on M Series routers, and on MX Series routers equipped with Multiservices DPCs. 6to4 tunnels are not supported on MX Series routers with MS-MPCs or MS-MICs.

To configure a 6to4 PMT:

1. Configure the ingress interface for 6to4 traffic. Include the name of the service set that identifies the rules for input and output service on this interface.

```
[edit interfaces ge-0/2/1]
user@host# set unit logical-unit-number family family service input service-set-name
user@host# set unit logical-unit-number family family service output service-set-name
user@host# set unit logical-unit-number family family address addres
```

   For example:

```
[edit interfaces ge-0/2/1]
user@host# set unit 0 family inet service input service-set v6to4-pmt
user@host# set unit 0 family inet service output service-set v6to4-pmt
user@host# set unit 0 family inet address 130.130.130.1/24
```

2. Configure the egress interface.

```
[edit interfaces ge-0/2/2]
user@host# set unit logical-unit-number family family address address
```

   For example:

```
[edit interfaces ge-0/2/2]
user@host# set unit 0 family inet6 address 4ABC::1/16
```

3. Configure the service interface that contains the rules for processing incoming traffic. Include a syslog option and associate a logical unit.

```
[edit interfaces sp-2/0/0]
user@host# edit services-options syslog host host-name services any
user@host# edit unit logical-unit-number family family
user@host# edit unit 0 family family
```

   For example:

```
[edit interfaces sp-2/0/0]
user@host# set services-options syslog host local services any
```

```
user@host# set unit 0 family inet
user@host# set unit 0 family inet6
```

4. Configure the softwire concentrator and softwire rule for 6to4. In the Junos OS, 6to4 PMT configuration uses the same options as 6rd.

   ```
   [edit services softwire softwire-concentrator v6rd v6to4]
   user@host# set softwire-address softwire-addres
   user@host# set ipv4-prefix ipv4-prefix
   user@host# set v6rd-prefix v6rd-prefix
   user@host# set mtu-v4 mtu-v4
   ```

   For example:

   ```
   [edit services softwire softwire-concentrator v6rd v6to4]
   user@host# set softwire-address 192.88.99.1
   user@host# set ipv4-prefix 130.130.130.2/32
   user@host# set v6rd-prefix 2002::0/16
   user@host# set mtu-v4 9192
   ```

5. Define the softwire rule that will process traffic on the ingress interface.

   ```
   [edit services softwire rule v6to4-r1]
   user@host# set match-direction input
   user@host# set term term-name then v6rd softwire-concentrator
   ```

   For example:

   ```
   [edit services softwire rule v6to4-r1]
   user@host# set match-direction input
   user@host# set term t1 then v6rd v6to4
   ```

6. Define a stateful firewall rule that will accept all incoming traffic on the ingress interface.

   ```
   [edit services stateful-firewall rule sfw-r1]
   user@host# set match-direction direction
   user@host# set term term-name then accept
   user@host# set term term-name then syslog
   ```

For example:

```
[edit services stateful-firewall rule sfw-r1]
user@host# set match-direction input-output
user@host# set term t1 then accept
user@host# set term t1 then syslog
```

7. Define the NAT pool to be used for IPv6 NAT translation. This pool supports translation of the Anycast 6to4 relay addresses to addresses at the provider-managed relay.

```
[edit services nat pool v6to4-pmt]
user@host# set address address
user@host# port automatic
```

For example:

```
[edit services nat pool v6to4-pmt]
user@host# set address 3ABC::1/128
user@host# set port automatic
```

8. Define the NAT rule for translation.

```
[edit services nat rule rule-name]
user@host# set match-direction input
user@host# set term term-name then translated source-pool pool-name
user@host# set term t1 then translated translation-type translation-type
```

For example:

```
[edit services nat rule v6to4-pmt-r1]
user@host# set match-direction input
user@host# set term t1 then translated source-pool v6to4-pmt
user@host# set term t1 then translated translation-type napt-66
```

9. Define the service set that specifies the softwire rule and NAT rule.

```
[edit services service-set v6to4-pmt]
user@host# set softwire-rules rule-name
user@host# set stateful-firewall-rules rule-name
```

```
user@host# set nat-rules rule-name
user@host# set interface-service service-interface interface-name
```

For example:

```
[edit services service-set v6to4-pmt]
user@host# set softwire-rules v6to4-r1
user@host# set stateful-firewall-rules sfw-r1
user@host# set nat-rules v6to4-pmt-r1
user@host# set interface-service service-interface sp-2/0/0
```

CHAPTER 21

Transition to IPv6 With DS-Lite Softwires

IN THIS CHAPTER

- DS-Lite Softwires | 407

DS-Lite Softwires

IN THIS SECTION

- Configuring a DS-Lite Softwire Concentrator | 407
- Configuring IPv6 Multicast Interfaces | 408
- Example: Basic DS-Lite Configuration | 409
- Example: Configuring DS-Lite and 6rd in the Same Service Set | 418
- DS-Lite Subnet Limitation | 427

Configuring a DS-Lite Softwire Concentrator

DS-Lite is supported on Multiservices 100, 400, and 500 PICs on M Series routers, and on MX Series routers equipped with Multiservices DPCs. Starting in Junos OS release 17.4R1, DS-Lite is supported on MX Series routers with MS-MPCs and MS-MICs.Starting in Junos OS release 19.2R1, DS-Lite is supported on MX Virtual Chassis and MX Broadband Network Gateway (BNG) routers.

To configure a DS-Lite softwire concentrator:

1. Assign a name to the DS-Lite softwire concentrator.

```
[edit services softwire softwire-concentrator]
user@host# edit ds-lite ds-lite-softwire-concentrator
```

2. Specify the address of the softwire tunnel.

```
[edit services softwire softwire-concentrator ds-lite ds-lite-softwire-concentrator]
user@host# set softwire-address address
```

3. Specify the MTU for the softwire tunnel.

```
[edit services softwire softwire-concentrator ds-lite ds-lite-softwire-concentrator]
user@host# set mtu-v6 bytes
```

NOTE: The `mtu-v6` option is supported on MX Series routers equipped with MS-DPCs. Starting in Junos OS release 18.1R1, the `mtu-v6` option is supported on MX Series routers with MS-MPCs or MS-MICs.

This option sets the maximum transmission unit when encapsulating IPv4 packets into IPv6. If the final length is greater than the MTU, the IPv6 packet will be fragmented. This option is mandatory since it depends on other network parameters under administrator control.

4. To copy DSCP information from the IPv6 header into the decapsulated IPv4 header, include the `copy-dscp` statement. This statement is not supported on MS-MPCs and MS-MICs.

```
[edit services softwire softwire-concentrator ds-lite ds-lite-softwire-concentrator]
user@host# set copy-dscp
```

5. Specify the maximum number of flows for the softwire.

```
[edit services softwire softwire-concentrator ds-lite ds-lite-softwire-concentrator]
user@host# set flow-limit 1000
```

SEE ALSO

Tunneling Services for IPv4-to-IPv6 Transition Overview | 394

Configuring IPv6 Multicast Interfaces

Configure multicast filters on Ethernet interfaces when IPv6 NAT is used for neighbor discovery. This enables the router to process softwire-initiated flows in both directions.

To configure IPv6 multicast interfaces:

1. Access the softwire hierarchy.

```
user@host# edit services softwire
```

2. Include the `ipv6-multicast-interfaces` statement for an individual interface.

```
[edit services softwire]
user@host# set ipv6-multicast-interfaces interface-name
```

   Or configure all softwire interfaces as IPv6 multicast.

```
[edit services softwire]
user@host# set ipv6-multicast-interfaces all
```

SEE ALSO

Tunneling Services for IPv4-to-IPv6 Transition Overview | 394

Example: Basic DS-Lite Configuration

IN THIS SECTION

- Requirements | 409
- Configuration Overview and Topology | 410
- Configuration | 411

DS-Lite employs IPv4-over-IPv6 tunnels to cross an IPv6 access network to reach a carrier-grade IPv4-IPv4 NAT. This facilitates the phased introduction of IPv6 on the Internet by providing backward compatibility with IPv4. See *Understanding IPv6 Dual-Stack Lite*.

Requirements

The following hardware components can perform DS-Lite:

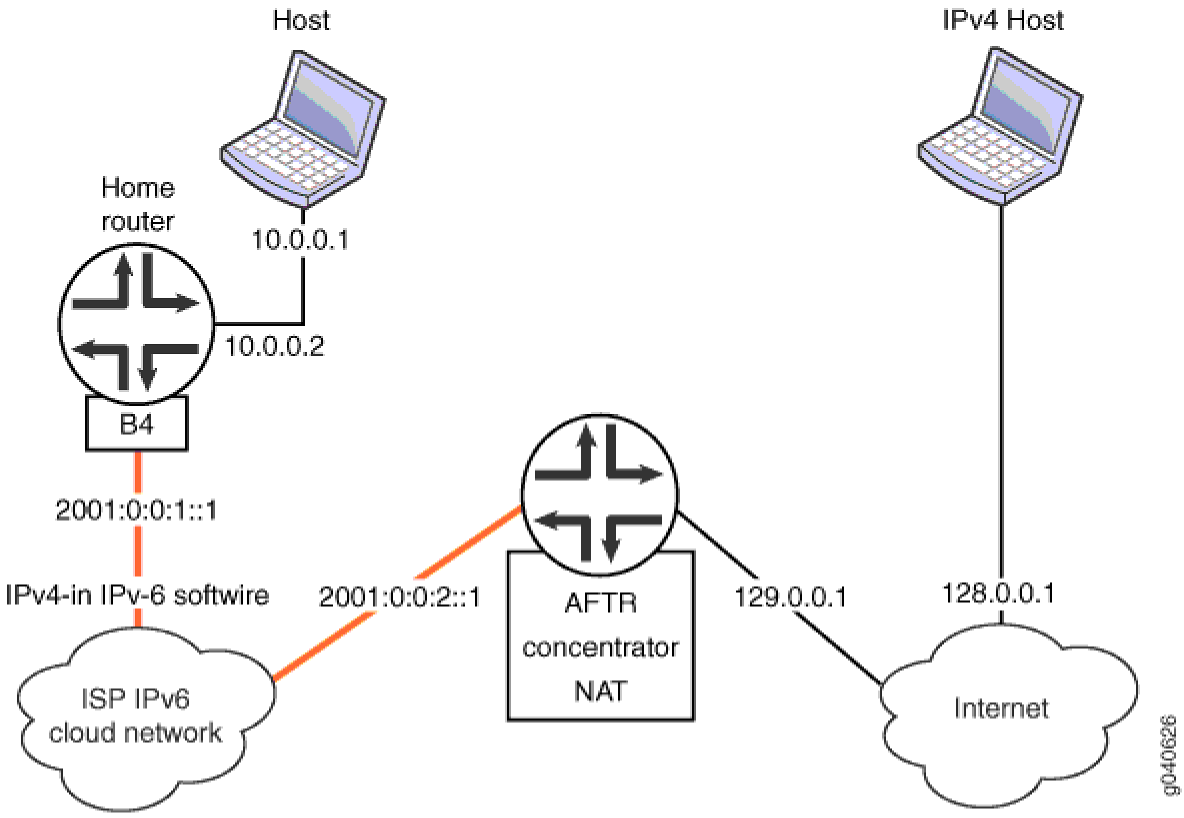
- M Series Multiservice Edge routers with Multiservices PICs.

- T Series Core routers with Multiservices PICs.
- MX Series 5G Universal Routing Platforms with Multiservices DPCs. Starting in Junos OS release 17.4R1, DS-Lite is supported on MX Series routers with MS-MPCs and MS-MICs.Starting in Junos OS release 19.2R1, DS-Lite is supported on MX Virtual Chassis and MX Broadband Network Gateway (BNG) routers.

Configuration Overview and Topology

This example describes how configure an MX Series router with an MS-DPC as an AFTR to facilitate the flow shown in Figure 31 on page 410 .

Figure 31: DS-Lite Topology



In this example, the DS-Lite softwire concentrator, or AFTR, is an MX Series router with two Gigabit interfaces and a Services DPC. The interface facing the B4 element is ge-3/1/5 and the interface facing the Internet is ge-3/1/0.

Configuration

IN THIS SECTION

- Chassis Configuration | 411
- Interfaces Configuration | 411
- Network Address and Port Translation Configuration | 413
- Softwire Configuration | 415
- Service Set Configuration | 416

Chassis Configuration

Step-by-Step Procedure

To configure the service PIC (FPC 0 Slot 0) with the Layer 3 service package:

1. Enter the **edit chassis** hierarchy level.

```
user@host# edit chassis
```

2. Configure the Layer 3 service package.

```
[edit chassis]
user@host# set fpc 0 pic 0 adaptive-services service-package layer-3
```

Interfaces Configuration

Step-by-Step Procedure

To configure interfaces facing the B4 (softwire initiator) and facing the Internet:

1. Go the `[edit interfaces]` edit hierachy level for ge-3/1/0, which faces the Internet.

```
host# edit interfaces ge-3/1/0
```

2. Define the interface.

```
[edit interfaces ge-3/1/0]
user@host# set description AFTR-Internet
user@host# set unit 0 family inet address 128.0.0.2/24
```

3. Go to the `[edit interfaces]` hierachy level for ge-3/1/5, which faces the B4.

```
user@host# up 1
[edit]
user@host# edit interfaces ge-3/1/5
```

4. Define the interface.

```
[edit interfaces ge-3/1/5]
user@host# set description AFTR-B4
user@host# set unit 0 family inet
user@host# edit unit 0 family inet6
[edit unit 0 family inet6]
user@host# set service input service-set sset
user@host# set service output service-set sset
user@host# set address 2001:0:0:2::1/48
```

5. Go to the `[edit interfaces]` hierarchy level for sp-0/0/0, used to host the DS-Lite AFTR.

```
[edit]
user@host# edit interfaces sp-0/0/0
```

6. Define the interface.

```
[edit interfaces sp-0/0/0]
user@host# set description AFTR-B4
user@host# set unit 0 family inet
user@host# edit unit 0 family inet6
```

Results

```
user@host# show interfaces ge-3/1/0
description AFTR-Internet;
unit 0 {
    family inet {
        address 128.0.0.2/24;
    }
}
user@host# show interfaces ge-3/1/5
description AFTR-B4;
unit 0 {
    family inet;
    family inet6 {
        service {
            input {
                service-set sset;
            }
            output {
                service-set sset;
            }
        }
        address 2001:0:0:2::1/48;
    }
}

user@host# show interfaces sp-o/o/o
unit 0 {
family inet;
family inet6;
}
```

Network Address and Port Translation Configuration

Step-by-Step Procedure

To configure NAPT:

1. Go to the `[edit services nat]` hierarchy level.

```
user@host# edit services nat
[edit services nat]
```

2. Define a NAT pool p1.

```
user@host# set pool p1 address 129.0.0.1/32 port automatic
```

3. Define a NAT rule, beginning with the match direction.

```
[edit services nat]
user@host# set rule r1 match-direction input
```

4. Define a **term** for the rule, beginning with a from clause.

```
[edit services nat]
user@host# set rule r1 term t1 from source-address 10.0.0.0/16
```

5. Define the desired translation in a **then** clause. In this case, use dynamic source translation.

```
[edit services nat]
user@host# set rule r1 term t1 then translated source-pool p1 translation-type napt-44
```

6. (Optional) Configure logging of translation information for the rule.

```
[edit services nat]
user@host# set rule r1 term t1 then syslog
```

Results

```
user@host# show services nat
pool p1 {
    address 129.0.0.1/32;
    port {
        automatic;
    }
}
rule r1 {
    match-direction input;
    term t1 {
        from {
            source-address {
                10.0.0.0/16;
            }
        }
        then {
            translated {
                source-pool p1;
                translation-type {
                    napt-44;
                }
            }
            syslog;
        }
    }
```

Softwire Configuration

Step-by-Step Procedure

To configure the DS-Lite softwire concentrator and associated rules:

1. Go to the `[edit services softwire]` hierarchy level.

   ```
   user@host# edit services softwire
   ```

2. Define the DS-Lite softwire concentrator.

   ```
   [edit services softwire]
   user@host# set softwire-concentrator ds-lite ds-1 softwire-address 1001::1 mtu-v6 1460
   ```

3. Define the softwire rule.

```
[edit services softwire]
user@host# set rule r1 match-direction input term t1 then ds-lite ds1.
```

Results

```
user@host# show services softwire
softwire-concentrator {
    ds-lite ds1 {
        softwire-address 1001::1;
        mtu-v6 1460;
    }
}
rule r1 {
    match-direction input;
    term t1 {
        then {
            ds-lite ds1;
        }
    }
}
```

Service Set Configuration

Step-by-Step Procedure

Configure a service set that includes softwire and NAT rules and specifies either interface-service or next-hop service. This example uses a next-hop service.

1. Go to the `[edit services service-set]` hierarchy level, naming the service set.

```
user@host# edit services service-set sset
```

2. Define the NAT rule to be used for IPv4-to-IPv4 translation.

```
[edit services service-set sset]
user@host# set nat-rules r1
```

3. Define the softwire rule to define the softwire tunnel.

```
[edit services service-set sset]
user@host# set softwire-rules r1
```

4. Define the interface service,

```
[edit services service-set sset]
user@host# set interface-service service-interface sp-0/0/0.0
```

TIP: In order to avoid or minimize IPv6 fragmentation, you can configure a TCP maximum segment size (MSS) for your service set.

5. (Optional) Define a TCP MSS.

```
[edit services service-set sset]
user@host# set tcp-mss 1024
```

Results

```
user@host# show services service-set
syslog {
    host local {
        services any;
    }
}
softwire-rules r1;
nat-rules r1;
interface-service {
    service-interface sp-0/0/0;
}
}
```

Example: Configuring DS-Lite and 6rd in the Same Service Set

IN THIS SECTION

- Requirements | 418
- Overview | 418
- Configuration | 418

Requirements

The following hardware components can perform DS-Lite:

- M Series Multiservice Edge routers with Multiservices PICs.
- T Series Core routers with Multiservices PICs.
- MX Series 5G Universal Routing Platforms with Multiservices DPCs. Starting in Junos OS release 17.4R1, DS-Lite is supported on MX Series routers with MS-MPCs and MS-MICs.Starting in Junos OS release 19.2R1, DS-Lite is supported on MX Virtual Chassis and MX Broadband Network Gateway (BNG) routers.Starting in Junos OS release 20.2R1, DS-Lite is supported for CGNAT Next Gen Services on MX240, MX480 and MX960 routers.

Overview

This example describes a softwire solution that includes DS-Lite and 6rd in the same service set.

Configuration

IN THIS SECTION

- Chassis Configuration | 419
- Softwire Concentrator, Softwire Rule, Stateful Firewall Rule Configuration | 421
- NAT Configuration for DS-Lite | 424
- Service Set Configuration | 425

Chassis Configuration

Step-by-Step Procedure

To configure the chassis:

1. Configure the ingress interface.

   ```
   user@host# edit interfaces ge-1/2/0
   [edit interfaces ge-1/2/0]
   user@host# set unit 0 family inet service input service-set v6rd-dslite-service-set
   user@host# set unit 0 family inet service output service-set v6rd-dslite-service-set
   user@host# set unit 0 family inet address address 10.10.10.1/24
   user@host# set unit 0 family inet6 service input service-set v6rd-dslite-service-set
   user@host# set unit 0 family inet6 service output service-set v6rd-dslite-service-set
   user@host# set unit 0 family inet6 address address address 2001::1/16
   ```

   Here the service set is applied on the inet (IPv4) and inet6 (IPv6) families of subunit 0. Both DS-Lite IPv6 traffic and 6rd IPv4 traffic hits the service filter and is sent to the services PIC.

2. Configure the egress interface (IPv6 Internet). The IPv4 server that the DS-Lite clients are trying to reach is at 200.200.200.2/24, and the IPv6 server is at 3ABC::2/16.

   ```
   user@host# edit interfaces ge-1/2/2
   [edit interfaces ge-1/2/2]
   user@host# set unit 0 family inet address 200.200.200.1/24
   user@host# set unit 0 family inet6 address 3ABC::1/16
   ```

3. Configure the services PIC.

   ```
   user@host# edit interfaces sp-3/0/0
   [edit interfaces sp-3/0/0]
   user@host# set unit 0 family inet
   user@host# set unit 0 family inet6
   ```

Results

```
[edit interfaces]
user@host# show
```

```
ge-1/2/0 {
    unit 0 {
        family inet {
            service {
                input {
                    service-set v6rd-dslite-service-set;
                }
                output {
                    service-set v6rd-dslite-service-set;
                }
            }
            address 10.10.10.1/24;
        }
        family inet6 {
            service {
                input {
                    service-set v6rd-dslite-service-set;
                }
                output {
                    service-set v6rd-dslite-service-set;
                }
            }
            address 2001::1/16;
        }
    }
}
ge-1/2/2 {
    unit 0 {
        family inet {
            address 200.200.200.1/24;
        }
        family inet6 {
            address 3ABC::1/16;
        }
    }
}
sp-3/0/0 {
    unit 0 {
        family inet;
        family inet6;
    }
}
```

Softwire Concentrator, Softwire Rule, Stateful Firewall Rule Configuration

Step-by-Step Procedure

To configure the softwire concentrator, softwire rule, and stateful firewall rule:

1. Configure the DS-Lite and 6rd softwire concentrators.

```
user@host# edit services softwire softwire-concentrator ds-lite ds1
[edit services softwire softwire-concentrator ds-lite ds1]
user@host# set softwire-address 1001::1
user@host# mtu-v6 9192
usert@host# up 1
usert@host# edit v6rd v6rd-dom1
[edit services softwire softwire-concentrator v6rd v6rd-dom1]
user@host# set softwire-address 30.30.30.1
user@host# set ipv4-prefix 10.10.10.0/24
user@host# set v6rd-prefix 3040::0/16
user@host# set mtu-v4 9192
```

2. Configure the softwire rules.

```
user@host# edit services softwire rule v6rd-r1]
[edit services softwire rule v6rd-r1]
user@host# set match-direction input
user@host# set term t1 then v6rd v6rd-dom1
user@host# up 1
user@host# edit services softwire]
[edit services softwire]
user@host# edit rule dslite-r1
[edit services softwire rule dslite-r1]
user@host# set term dslite-t1 then ds-lite ds1
```

The following routes are added by the services PIC daemon on the Routing Engine:

```
user@host# run show route 30.30.30.1
inet.0: 43 destinations, 46 routes (42 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

30.30.30.1/32      *[Static/786432] 00:24:11
                      Service to v6rd-dslite-service-set
```

```
[edit]
user@host# run show route 3040::0/16

inet6.0: 23 destinations, 33 routes (23 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

3040::/16          *[Static/786432] 00:24:39
                      Service to v6rd-dslite-service-set
```

```
user@host# run show route 1001::1
inet6.0: 33 destinations, 43 routes (33 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1001::1/128        *[Static/1] 1w2d 22:05:41
                      Service to v6rd-dslite-service-set
```

3. Configure a stateful firewall rule.

```
user@host# edit services stateful-firewall rule r1
[edit services stateful-firewall rule r1]
user@host# set match-direction input-output
user@host# set term t1 then accept
```

```
[edit services stateful-firewall]
    rule r1 {
        match-direction input-output;
        term t1 {
            then {
                accept;
            }
        }
    }
```

Results

```
[edit services softwire]
user@host# show
softwire-concentrator {
    ds-lite ds1 {
        softwire-address 1001::1;
        mtu-v6 9192;
    }
    v6rd v6rd-dom1 {
        softwire-address 30.30.30.1;
        ipv4-prefix 10.10.10.0/24;
        v6rd-prefix 3040::0/16;
        mtu-v4 9192;
    }
}
rule v6rd-r1 {
    match-direction input;
    term t1 {
        then {
            v6rd v6rd-dom1;
        }
    }
}
rule dslite-r1 {
    match-direction input;
    term dslite-t1 {
        then {
            ds-lite ds1;
        }
    }
}
```

```
[edit services stateful-firewall]
user@host# show
rule r1 {
    match-direction input-output;
    term t1 {
        then {
            accept;
        }
    }
}
```

NAT Configuration for DS-Lite

Step-by-Step Procedure

To configure NAT for DS-Lite:

1. Configure a NAT pool for DS-Lite.

   ```
   user@host# edit services nat pool dslite-pool
   [edit services nat pool dslite-pool]
   user@host# set address-range low 33.33.33.1 high 33.33.33.32
   user@host# set port automatic
   ```

2. Configure a NAT rule.

   ```
   user@host# up 1
   [edit services nat rule dslite-nat-r1]
   user@host# set match-direction input
   user@host# set term dslite-nat-t1 from source-address 20.20.0.0/16 then translated
   translation-type napt-44
   ```

Results

```
[edit services nat]
user@host# show
pool dslite-pool {
    address-range low 33.33.33.1 high 33.33.33.32;
    port {
        automatic;
    }
}
rule dslite-nat-r1 {
    match-direction input;
    term dslite-nat-t1 {
        from {
            source-address {
```

```
                20.20.0.0/16;
            }
        }
        then {
            translated {
                source-pool dslite-pool;
                translation-type {
                    source dynamic;
                }
            }
        }
    }
}
```

Because of this NAT rule, the following NAT routes are installed for the reverse DS-Lite traffic:

```
user@host# run show route 33.33.33.0/24
inet.0: 48 destinations, 52 routes (47 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

33.33.33.1/32      *[Static/1] 1w2d 23:08:38
                      Service to v6rd-dslite-service-set
33.33.33.2/31      *[Static/1] 1w2d 23:08:38
                      Service to v6rd-dslite-service-set
33.33.33.4/30      *[Static/1] 1w2d 23:08:38
                      Service to v6rd-dslite-service-set
33.33.33.8/29      *[Static/1] 1w2d 23:08:38
                      Service to v6rd-dslite-service-set
33.33.33.16/28     *[Static/1] 1w2d 23:08:38
                      Service to v6rd-dslite-service-set
33.33.33.32/32     *[Static/1] 1w2d 23:08:38
                      Service to v6rd-dslite-service-set
```

The NAT rule triggers address translation for the traffic coming from 20.20.0.0/16 to public address range 33.33.33.1 to 33.33.33.32.

Service Set Configuration

Step-by-Step Procedure

This service set has a stateful firewall rule and 6rd rule for 6rd service. The service set also includes a softwire rule for DS-Lite and a NAT rule to perform address translation for all DS-Lite traffic. The NAT rule performs NAPT translation in the forward direction on the source address and port of the DS-Lite traffic.

To configure the service set:

1. Define the service set.

```
user@host# edit services service-set v6rd-dslite-service-set
```

2. Configure the service set rules.

```
[edit services service-set v6rd-dslite-service-set]
user@host# set softwire-rules dslite-r1
user@host# set stateful-firewall-rules r1
user@host# set nat-rules dslite-nat-r1
```

3. Configure the service set interface-service.

```
[edit services service-set v6rd-dslite-service-set]
user@host# set interface-service service-interface sp-3/0/0
```

Results

```
[edit services service-set]
user@host# show
v6rd-dslite-service-set {
    softwire-rules v6rd-r1;
    softwire-rules dslite-r1;
    stateful-firewall-rules r1;
    nat-rules dslite-nat-r1;
    interface-service {
        service-interface sp-3/0/0;
    }
```

SEE ALSO

Configuring Service Sets for Software | 400

DS-Lite Subnet Limitation

IN THIS SECTION

- DS-Lite Per Subnet Limitation Overview | 427
- Configuring DS-Lite Per Subnet Session Limitation to Prevent Denial of Service Attacks | 429

DS-Lite Per Subnet Limitation Overview

Junos OS enables you to limit the number of softwire flows from a subscriber's basic bridging broadband (B4) device at a given point in time, preventing subscribers from excessive use of addresses within the subnet. This limitation reduces the risk of denial-of-service (DoS) attacks. This limitation is supported on MX Series routers equipped with MS-DPCs. Starting in Junos OS Release 18.2R1, MS-MPCs and MS-MICs also support the subnet limitation feature.Starting in Junos OS Release 19.2R1, MX Virtual Chassis and MX Broadband Network Gateway (BNG) routers also support the subnet limitation feature.Starting in Junos OS release 20.2R1, DS-Lite is supported for CGNAT Next Gen Services on MX240, MX480 and MX960 routers.

A household using IPv6 with DS-Lite is a subnet, not just an individual IP address. The subnet limitation feature associates a subscriber and mapping with an IPv6 prefix instead of an IPv6 address. A subscriber can use any IPv6 addresses in that prefix as a DS-Lite B4 address and potentially exhaust carrier-grade NAT resources. The subnet limitation feature enables greater control of resource utilization by identifying a subscriber with a prefix instead of a specific address.

The subnet limit provides the following features:

- Flows utilize the complete B4 address.
- Prefix length can be configured per service set under softwire-options for the individual service-set.
- Port blocks are allocated per prefix of the subscriber B4 device, and not on each B4 address (if the prefix length is less than 128). If the prefix length is 128, then each IPv6 address is treated as a B4. Port blocks are allocated per 128-bit IPv6 address.
- Session limit, defined under the DS-Lite softwire concentrator configuration, limits the number of IPv4 sessions for the prefix.
- EIM, EIF, and PCP mappings are created per softwire tunnel (full 128 bit IPv6 address). Stale mappings time out based on timeout values.
- If prefix length is configured , then PCP `max-mappings-per-subscriber` (configurable under `pcp-server`) is based on the prefix only, and not the full B4 address.

- SYSLOGS for PBA allocation and release contain the prefix portion of the address completed with all zeros. SYSLOGS for PCP allocate and release, flow creation and deletion will still contain the complete IPv6 address.

The `show services nat mappings address-pooling-paired` operational command output now shows the mapping for the prefix. The mapping shows the address of the active B4.

The `show services softwire statistics ds-lite` output includes a new field that displays the number of times the session limit was exceeded for the MPC.

For Next Gen Services on MX240, MX480, and MX960 routers, the subnet limit statistic is displayed in the `Softwire session limit exceeded` field.

show services softwire statistics (MX-SPC3)

```
user@host> show services softwire statistics
vms-2/0/0
    Total Session Interest events                :3
    Total Session Destroy events                            :2
    Total Session Public Request events                     :0
    Total Session Accepts                                   :1
    Total Session Discards                                  :0
    Total Session Ignores                                   :0
    Total Session extension alloc failures                  :0
    Total Session extension set failures                    :0
Softwire statistics
    Total Softwire sessions created                         :1
    Total Softwire sessions deleted                         :2
    Total Softwire sessions created for reverse packets     :1
    Total Softwire session create failed for reverse pkts   :0
    Total Softwire rule match success                       :1
    Total Softwire rule match failed                        :0
    Softwire session limit exceeded                         :0
Softwire packet statistics
    Total Packets processed                                 :1
    Total packets encapsulated                              :1
    Total packets decapsulated                              :1
    Encapsulation errors                                    :0
    Decapsulation errors                                    :0
    Encapsulated pkts re-inject failures                    :0
    Decapsulated pkts re-inject failures                    :0
    DS-Lite ICMPv4 Echo replies sent                        :0
    DS-Lite ICMPv4 TTL exceeded messages sent               :0
    ICMPv6 ECHO request messages received destined to AFTR  :0
    ICMPv6 ECHO reply messages sent from AFTR              :0
    ICMPv6 ECHO requests to AFTR process failures          :0
    V6 untunnelled packets destined to AFTR dropped        :1
    Softwire policy add errors                             :0
    Softwire policy delete errors                          :0
    Softwire policy memory alloc failures                  :0
    Softwire Untunnelled packets ignored                   :0
Softwire Misc errors
    DS-Lite ICMPv4 TTL exceed message process errors       :0
```

SEE ALSO

show services nat source mappings address-pooling-paired

show services softwire statistics

Configuring DS-Lite Per Subnet Session Limitation to Prevent Denial of Service Attacks

You can configure the DS-Lite per subnet limitation on MX Series routers equipped with MS-DPCs. Starting in Junos OS Release 18.2R1, MS-MPCs and MS-MICs also support the subnet limitation feature. Starting in Junos OS Release 20.2R1, the Next Gen Services MX-SPC3 security services card supports the subnet limitation feature.

Starting in Junos OS Release 19.2R1, MX Virtual Chassis and MX Broadband Network Gateway (BNG) routers also support the subnet limitation feature.

To configure DS-Lite per subnet session limitation:

1. Configure the size of the subnet prefix to which limiting is applied. Specify a prefix length of 56, 64, 96, or 128.

   ```
   [edit}
   user@host# set services service-set service-set-name softwire-options dslite-ipv6-prefix-length dslite-ipv6-prefix-length
   ```

   NOTE: Ensure that all mappings are cleared before changing the prefix length.

2. If you are using a next-hop service set on an AMS interface for DS-Lite, set the AMS inside interface's IPv6 source prefix length to the same value you use for the subnet prefix in Step 1.

```
[edit interfaces interface-name unit interface-unit-number load-balancing-options hash-keys]
user@host# set ipv6-source-prefix-length ipv6-source-prefix-length
```

3. Configure the maximum number of subscriber sessions allowed per prefix. You can configure from 0 through 16,384 sessions.

```
[edit}
user@host# set services softwire softwire-concentrator dslite dslite-concentrator-name
session-limit-per-prefix 12
```

For Next Gen Services DS-Lite, MAP-E and V6rd softwires, configure the maximum number of subscriber sessions allowed per prefix:

```
[edit}
user@host# set services softwires softwire-types ds-lite | map-e | v6rd session-limit-per-
prefix limit
```

NOTE: You cannot use `flow-limit` and `session-limit-per-prefix` in the same `dslite` configuration.

SEE ALSO

clear services nat mappings

softwire-options

ds-lite

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
20.2R1	Starting in Junos OS release 20.2R1, DS-Lite is supported for CGNAT Next Gen Services on MX240, MX480 and MX960 routers.

20.2R1	Starting in Junos OS release 20.2R1, DS-Lite is supported for CGNAT Next Gen Services on MX240, MX480 and MX960 routers.
20.2R1	Starting in Junos OS Release 20.2R1, the Next Gen Services MX-SPC3 security services card supports the subnet limitation feature.
19.2R1	Starting in Junos OS release 19.2R1, DS-Lite is supported on MX Virtual Chassis and MX Broadband Network Gateway (BNG) routers.
19.2R1	Starting in Junos OS release 19.2R1, DS-Lite is supported on MX Virtual Chassis and MX Broadband Network Gateway (BNG) routers.
19.2R1	Starting in Junos OS release 19.2R1, DS-Lite is supported on MX Virtual Chassis and MX Broadband Network Gateway (BNG) routers.
19.2R1	Starting in Junos OS Release 19.2R1, MX Virtual Chassis and MX Broadband Network Gateway (BNG) routers also support the subnet limitation feature.
18.2R1	Starting in Junos OS Release 18.2R1, MS-MPCs and MS-MICs also support the subnet limitation feature.
18.1R1	Starting in Junos OS release 18.1R1, the `mtu-v6` option is supported on MX Series routers with MS-MPCs or MS-MICs.
17.4R1	Starting in Junos OS release 17.4R1, DS-Lite is supported on MX Series routers with MS-MPCs and MS-MICs.
17.4R1	Starting in Junos OS release 17.4R1, DS-Lite is supported on MX Series routers with MS-MPCs and MS-MICs.
17.4R1	Starting in Junos OS release 17.4R1, DS-Lite is supported on MX Series routers with MS-MPCs and MS-MICs.

CHAPTER 22

Transition to IPv6 With 6rd Softwires

IN THIS CHAPTER

- Configuring a 6rd Softwire | 432

Configuring a 6rd Softwire

IN THIS SECTION

- Configuring a 6rd Softwire Concentrator | 432
- Configuring Stateful Firewall Rules for 6rd Softwire | 433
- Example: Basic 6rd Configuration | 435
- High Availability and Load Balancing for 6rd Softwires | 442

Configuring a 6rd Softwire Concentrator

The 6rd feature is supported on Multiservices 100, 400, and 500 PICs on M Series routers, and on MX Series routers equipped with Multiservices DPCs. The 6rd feature is not supported on MX Series routers with MS-MPCs or MS-MICs.

To configure a 6rd softwire concentrator:

1. Assign a name to the 6rd softwire concentrator.

```
[edit services softwire softwire-concentrator]
user@host# edit v6rd v6rd-softwire-concentator
```

2. Specify the address of the softwire tunnel.

```
[edit services softwire softwire-concentrator v6rd v6rd-softwire-concentator]
user@host# set softwire-address address
```

3. Specify the MTU for the softwire tunnel.

```
[edit services softwire softwire-concentrator v6rd v6rd-softwire-concentator]
user@host# set mtu-v4 mtu-v4
```

TIP: In this release there is no support for fragmentation and reassembly, therefore the MTUs on the IPv6 and IPV4 network must be properly configured by the administrator.

NOTE: Configuration changes to 6rd softwire concentrators do not become effective in the Packet Forwarding Engine. This is a known limitation. If you attempt to add the new configuration of softwire concentrators by overriding the existing configuration of 1024 softwire concentrators, which is the maximum limit of softwire concentrators that the system supports, the new configuration is not updated. To work around this limitation, you must delete the existing configuration and commit the settings, and then add the new configuration of softwire concentrators and commit the settings.

NOTE: For 6rd softwire concentrators, packet drops are observed and error messages logged on the virtual terminal session (VTY) console, if one inline services (`si-`) interface is replaced with another `si-` interface without stopping the traffic during the replacement of the interface. In a scenario in which an `si-` interface is associated with a service set that has a large number of softwire concentrators, replacing that interface without halting the traffic causes traffic disruption. You must stop the traffic and restart it during such a replacement of `si-` interfaces with 6rd softwire concentrators. The following error messages are displayed on the VTY console of the FPC:

```
packet discarded because no ifl or not SI ifl
```

Configuring Stateful Firewall Rules for 6rd Software

You must configure a stateful firewall rule for use with 6rd softwires. The stateful firewall service is used only to direct packets to the softwire, not for firewalling purposes. The 6rd softwire service itself must be stateless. To support stateless processing, you must include an **allow** term in both directions of the stateful firewall policy.

The 6rd feature is supported on Multiservices 100, 400, and 500 PICs on M Series routers, and on MX Series routers equipped with Multiservices DPCs. The 6rd feature is not supported on MX Series routers with MS-MPCs or MS-MICs.

To include a stateful firewall rule for 6rd softwire processing:

1. Assign a name to the rule.

```
[edit services stateful-firewall]
user@host# edit rule rule-name
```

2. Specify the match direction.

```
[edit services stateful-firewall rule-name]
user@host# set match-direction input-output
```

3. Assign a name for the term.

```
[edit services stateful-firewall rule-name]
user@host# edit term term-name
```

4. Specify that all traffic in both directions should be accepted for the softwire process.

```
[edit services stateful-firewall rule-name term term-name]
user@host# set then accept
```

SEE ALSO

Tunneling Services for IPv4-to-IPv6 Transition Overview | **394**

Example: Configuring DS-Lite and 6rd in the Same Service Set | **418**

Example: Basic 6rd Configuration

IN THIS SECTION

- Requirements | 435
- Overview | 435
- Configuration | 435

Requirements

NOTE: The 6rd feature is supported on Multiservices 100, 400, and 500 PICs on M Series routers, and on MX Series routers equipped with Multiservices DPCs. The 6rd feature is not supported on MX Series routers with MS-MPCs or MS-MICs.

This example describes how a 6rd concentrator can be configured for a 6rd domain, D1, to provide IPv6 Internet connectivity.

The following hardware components can perform 6rd:

- M Series Multiservice Edge routers with Multiservices PICs
- T Series Core routers with Multiservices PICs
- MX Series 5G Universal Routing Platforms with Multiservices DPCs

Overview

This configuration example describes how to configure a basic 6rd tunneling solution.

Configuration

IN THIS SECTION

- CLI Quick Configuration | 436
- Chassis Configuration | 436
- Softwire Concentrator, Softwire Rule, and Stateful Firewall Rule Configuration | 439

- Service Set Configuration | **441**

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the `[edit]` hierarchy level.

```
set interfaces ge-1/2/0 unit 0 family inet service input service-set v6rd-dom1-service-set
set interfaces ge-1/2/0 unit 0 family inet service output  service-set v6rd-dom1-service-set
set interfaces ge-1/2/0 unit 0 family inet address 10.10.10.1/24
set interfaces ge-1/2/0 unit 0 family inet6 service input service-set v6rd-dom1-service-set
set interfaces ge-1/2/0 unit 0 family inet6 service output service-set v6rd-dom1-service-set
set interfaces ge-1/2/2 unit 0 family inet6 address 3abc::1/16
set interfaces sp-0/2/0 unit 0 family inet
set interfaces sp-0/2/0 unit 0 family inet6
set services  softwire softwire-concentrator v6rd v6rd-dom1 softwire-address 30.30.30.1
set services softwire softwire-concentrator v6rd v6rd-dom1 ipv4-prefix 10.10.10.0/24
set services softwire softwire-concentrator v6rd v6rd-dom1 v6rd-prefix 3040::0/16
set services softwire softwire-concentrator v6rd v6rd-dom1 mtu-v4 9192
set services softwire rule v6rd-dom1 match-direction input
set services softwire rule v6rd-dom1 term t1 then v6rd v6rd-dom1
set services service-set v6rd-dom1-service-set softwire-rules v6rd-dom1
set services service-set v6rd-dom1-service-set stateful-firewall-rules r1
set services service-set v6rd-dom1-service-set interface-service service-interface sp-0/2/0
set services stateful-firewall rule r1 match-direction input-output
set services stateful-firewall rule r1 term t1 then accept
```

Chassis Configuration

Step-by-Step Procedure

To configure the chassis:

1. Define the ingress interface.

   ```
   user@host# edit interfaces ge-1/2/0
   ```

2. Configure the ingress interface logical unit and input/output service options.

```
[edit interfaces ge-1/2/0]
user@host# set unit 0 family inet service input service-set v6rd-dom1-service-set
user@host# set unit 0 family inet service output service-set v6rd-dom1-service-set
user@host# set unit 0 family inet6 service input service-set v6rd-dom1-service-set
user@host# set unit 0 family inet6 service output service-set v6rd-dom1-service-set
```

3. Configure the address of the ingress interface.

```
[edit interfaces ge-1/2/0]
user@host# set unit 0 family inet address 10.10.10.1/24
```

4. Define the egress interface.

```
user@host# up
[edit interfaces]
user@host# edit ge-1/2/2
```

5. Define the logical unit and address for the egress interface.

```
[edit interfaces ge-1/2/2]
user@host# set unit 0 family inet6 address 3ABC::1/16
```

6. Define the services PIC.

```
[edit interfaces ge-1/2/2]
user@host# up
[edit interfaces]
user@host# edit sp-0/2/0
```

7. Configure the logical unit for the services PIC.

```
[edit interfaces sp-0/2/0]
user@host# up
[edit interfaces]
```

```
user@host# set unit 0 family inet
user@host# set unit 0 family inet6
```

Results

```
[edit interfaces]
user@host# show
sp-0/2/0 {
    unit 0 {
        family inet;
        family inet6;
    }
}
ge-1/2/0 {
    unit 0 {
        family inet {
            service {
                input {
                    service-set v6rd-dom1-service-set;
                }
                output {
                    service-set v6rd-dom1-service-set;
                }
            }
            address 10.10.10.1/24;
        }
        family inet6 {
            service {
                input {
                    service-set v6rd-dom1-service-set;
                }
                output {
                    service-set v6rd-dom1-service-set;
                }
            }
        }
    }
}
ge-1/2/2 {
    unit 0 {
        family inet6 {
            address 3abc::1/16;
        }
    }
}
```

Softwire Concentrator, Softwire Rule, and Stateful Firewall Rule Configuration

Step-by-Step Procedure

To configure the softwire concentrator, softwire rule, and stateful firewall rule:

1. Define the 6rd softwire concentrator.

   ```
   user@host# top
   user@host# edit services softwire softwire-concentrator v6rd v6rd-dom1
   ```

2. Configure the softwire concentrator properties. Here, softwire address 30.30.30.1 is the softwire concentrator IPv4 address, 10.10.10.0/24 is the IPv4 prefix of the CE WAN side, and 3040::0/16 is the IPv6 prefix of the 6rd domain D1.

   ```
   [edit services softwire softwire-concentrator v6rd v6rd-dom1]
   user@host# set softwire-address 30.30.30.1
   user@host# set ipv4-prefix 10.10.10.0/24
   user@host# set v6rd-prefix 3040::0/16
   user@host# set mtu-v4 9192
   ```

3. Define the softwire rule.

   ```
   [edit services softwire softwire-concentrator v6rd v6rd-dom1]
   user@host# up 2
   [edit services softwire]
   user@host# edit rule v6rd-dom1
   [edit services softwire rule v6rd-dom1]
   user@host# set match-direction input
   [edit services softwire rule v6rd-dom1]
   user@host# set term t1 then v6rd v6rd-dom1
   ```

4. Define a stateful firewall rule and properties. You must configure a stateful firewall rule that accepts all traffic in both the input and output direction in order for 6rd to work; however, this is not

enforced through the CLI. This is because in IPv6, gratuitous IPv6 packets are expected (due to Anycast) and should not be dropped. The service PIC can handle reverse traffic without seeing all forward traffic. This can also happen with service PIC switchover in the middle of a session. By default, the stateful firewall on the service PIC will drop all traffic unless a rule is configured explicitly to allow it.

```
[edit services softwire softwire-concentrator v6rd v6rd-dom1]
user@host# up 3
[edit servicesl]
user@host# edit services stateful-firewall
[edit services stateful-firewall]
user@host# edit rule r1
[edit services stateful-firewall rule r1]
user@host# set match-direction input-output
user@host# set term t1 then accept
```

Results

```
[edit services softwire]
user@host# show
softwire-concentrator {
    v6rd v6rd-dom1 {
        softwire-address 30.30.30.1;
        ipv4-prefix 10.10.10.0/24;
        v6rd-prefix 3040::0/16;
        mtu-v4 9192;
    }
}
rule v6rd-dom1-r1 {
    match-direction input;
    term t1 {
        then {
            v6rd v6rd-dom1;
        }
    }
}
```

Service Set Configuration

Step-by-Step Procedure

To configure the service set:

1. Define the service set for 6rd processing.

```
user@host# top
user@host# edit services service-set v6rd-dom1-service-set
```

2. Define the softwire and stateful firewall rules for the service set.

```
[edit services service-set v6rd-dom1-service-set]
user@host# set softwire-rules v6rd-dom1
user@host# set stateful-firewall-rules r1
```

3. Define the interface-service for the service set.

```
[edit services service-set v6rd-dom1-service-set]
user@host# set interface-service service-interface sp-0/2/0
```

Results

```
[edit service-set v6rd-dom1-service-set]
user@host# show
softwire-rules v6rd-dom1-r1
    interface-service {
        service-interface sp-0/2/0;
    }
```

SEE ALSO

Tunneling Services for IPv4-to-IPv6 Transition Overview | **394**

Example: Basic DS-Lite Configuration | **409**

Example: Configuring DS-Lite and 6rd in the Same Service Set | **418**

High Availability and Load Balancing for 6rd Softwires

IN THIS SECTION

- Load Balancing a 6rd Domain Across Multiple Services PICs | 442
- Example: Load Balancing a 6rd Domain Across Multiple Services PICs | 442
- Configuring High Availability for 6rd Using 6rd Anycast | 450

NOTE: The 6rd feature is supported on Multiservices 100, 400, and 500 PICs on M Series routers, and on MX Series routers equipped with Multiservices DPCs. The 6rd feature is not supported on MX Series routers with MS-MPCs or MS-MICs.

Load Balancing a 6rd Domain Across Multiple Services PICs

The 6rd domain is an IPv6 network, which can potentially be very large. A single PIC, or network processing unit (NPU) on a Multiservices DPC, might not be able to handle all the traffic for the 6rd domain. To alleviate load problems, you can load-balance the 6rd domain traffic across multiple PICs. To do so, assign the same softwire rule to different services sets that use different interfaces. Configure explicit routes and equal-cost multipath (ECMP) to load-balance the 6rd traffic.

Example: Load Balancing a 6rd Domain Across Multiple Services PICs

IN THIS SECTION

- Hardware and Software Requirements | 442
- Overview | 443
- Configuration | 443

Hardware and Software Requirements

This example requires the following hardware:

- An MX Series 5G Universal Routing Platform with a services DPC with two available NPUs or an M Series Multiservice Edge router with two services PICs available for 6rd softwire concentrator processing
- A domain name server (DNS)

This example uses the following software:

- Junos OS Release 11.4 or higher

Overview

Because of anticipated volume, a provider needs to balance 6rd softwire traffic between two services PICs.

Configuration

IN THIS SECTION

- Chassis Configuration | 443
- Softwire Concentrator and Softwire Rule Configuration | 444
- Stateful Firewall Configuration | 446
- Service Set Configuration | 446
- Load-Balancing Configuration | 447

Chassis Configuration

Step-by-Step Procedure

To configure the chassis:

1. Define the ingress interface and its properties.

```
user@host# edit interfaces ge-1/2/0
user@host# set unit 0 family inet address 10.10.10.1/16
```

2. Define the egress interface and its properties. In this example, the IPv6 clients try to reach the IPv6 server at 3abc::2/16.

   ```
   user@host# edit interfaces ge-1/2/2
   user@host# set unit 0 family inet6 address 3ABC::1/16
   ```

3. Define the services PICs for selection as softwire concentrators by the load-balancing process. This configuration uses two PICs/NPUs: sp-3/0/0 and sp-3/1/0. A next-hop style service set is configured (shown in the next section).

   ```
   user@host# edit interfaces sp-3/0/0
   [edit interfaces ge-3/0/0]
   user@host# set services-options syslog host local services any
   user@host# set unit 0 family inet
   user@host# set unit 0 family inet6
   user@host# set unit 1 family inet service-domain inside
   user@host# set unit 1 family inet service-domain outside
   user@host# set unit 2 family inet service-domain inside
   user@host# set unit 2 family inet service-domain outside
   user@host# up 1
   [edit]
   user@host# edit interfaces sp-3/1/0
   [edit interfaces sp-3/1/0]
   user@host# set services-options syslog host local services any
   user@host# set unit 0 family inet
   user@host# set unit 0 family inet6
   user@host# set unit 1 family inet service-domain inside
   user@host# set unit 1 family inet service-domain outside
   user@host# set unit 2 family inet service-domain inside
   user@host# set unit 2 family inet service-domain outside
   ```

Softwire Concentrator and Softwire Rule Configuration

Step-by-Step Procedure

The softwire configuration is straightforward. In this example, the 6rd domain prefix is 3040::0/16, the 6rd softwire concentrator IPv4 address is 30.30.30.1, and the customer IPv4 network is 10.10.0.0/16. In the customer premises equipment (CPE) network, all customer edge (CE) devices have addresses that belong to the 10.10.0.0/16 network. To configure the softwire:

1. Go to the `[edit services softwire]` hierarchy level.

   ```
   user@host# edit services softwire
   ```

2. Configure IPv6 multicast.

   ```
   [edit services softwire]
   user@host# set ipv6-multicast-interfaces all
   ```

3. Go to the softtwire concentrator v6rd hierarchy level and name the softwire concentrator **shenick01-rd1**.

   ```
   [edit services softwire]
   user@host# edit softwire-concentrator v6rd shenick01-rd1
   ```

4. Configure the softwire concentrator properties.

   ```
   [edit services softwire softwire-concentrator v6rdshenick01-rd1 ]
   user@host# set softwire-address 30.30.30.1
   user@host# set ipv4-prefix 10.10.0.0/16
   user@host# set v6rd-prefix 3040::/16
   user@host# set mtu-v4 9192
   ```

5. Configure a softwire rule for incoming 6rd traffic.

   ```
   [edit services softwire softwire-concentrator v6rd shenick01-rd1 ]
   user@host# up 1
   [edit services softwire ]
   user@host# edit rule shenick01-r1
   [edit services softwire rule shenick01-r1]
   user@host# set match-direction input
   user@host# set term t1 then v6rd shenick01-rd1
   ```

Stateful Firewall Configuration

Step-by-Step Procedure

To configure the stateful firewall rule:

1. Go to the stateful firewall hierarchy level and define a rule.

```
user@host# edit services stateful-firewall rule r1
```

2. Set the match direction.

```
[edit services stateful-firewall rule r1]
user@host# set match-direction input-output
```

3. Configure a term that accepts all traffic.

```
[edit services stateful-firewall rule r1]
user@host# set term t1 then accept
```

Service Set Configuration

Step-by-Step Procedure

This configuration provides two service sets, each pointing to a different network processing unit (NPU). Both service sets use the same stateful firewall and softwire rules. Because they use the same softwire rule, they refer to same 6rd softwire concentrator. This results in the software concentrator being hosted on both the NPUs.

To configure the service set:

1. Define a service set for the first NPU.

```
user@host# edit services service-set v6rd-sset1
```

2. Configure the softwire and stateful firewall rules for the first NPU.

```
[edit services service-set v6rd-sset1]
user@host# set softwire-rules shenick01-r1
user@host# set stateful-firewall-rules r1
```

3. Configure the inside and outside interfaces for the next-hop service.

```
[edit services service-set v6rd-sset1]
user@host# set next-hop-service inside-service-interface sp-3/0/0.1
user@host# set next-hop-service outside-service-interface sp-3/0/0.2
```

4. Define a service set for the second NPU.

```
user@host# edit services service-set v6rd-sset2
```

5. Configure the softwire and stateful firewall rules for the second NPU.

```
[edit services service-set v6rd-sset2]
user@host# set softwire-rules shenick01-r1
user@host# set stateful-firewall-rules r1
```

6. Configure the inside and outside interfaces for the next-hop service.

```
[edit services service-set v6rd-sset1]
user@host# set next-hop-service inside-service-interface sp-3/1/0.1
user@host# set next-hop-service outside-service-interface sp-3/1/0.2
```

Load-Balancing Configuration

Step-by-Step Procedure

To configure load balancing:

Configure explicit routes and ECMP to load-balance the 6rd traffic. Configure explicit routes for both the 6rd concentrator IPv4 address and the 6rd domain prefix, so that they point to both NPUs.

1. To configure static routes for the 6rd domain using the routing-table inet6.0, go to the `[edit forwarding-options rib inet6.0 static]` hierarchy level and set the routes for the 6rd domain and the 6rd concentrator IPv4 address.

   ```
   user@host edit forwarding-options rib inet6.0 static
   [edit forwarding-options rib inet6.0 static]
   user@host# set route 3040::0/16 next-hop [ sp-3/0/0.2 sp-3/1/0.2 ]
   user@host# set route 30.30.30.1/32 next-hop [ sp-3/0/0.1 sp-3/1/0.1 ]
   ```

   The service PIC daemon (spd) also adds default routes to these addresses pointing to the NPUs. However, the routes added by the spd use different metrics, which are computed based on the FPC, PIC, slot numbers, and subunit of the services PIC if used in the service set configuration. The static routes configured in this sample configuration will have metrics of 5 and therefore a higher preference than the spd-added routes.

   The explicitly configured routes are as follows:

   ```
   root@host# run show route 30.30.30.1
   inet.0: 37 destinations, 40 routes (36 active, 0 holddown, 1 hidden)
   + = Active Route, - = Last Active, * = Both

   30.30.30.1/32      *[Static/5] 00:00:10
                       > via sp-3/0/0.1
                         via sp-3/1/0.1
                       [Static/786433] 00:23:03
                       > via sp-3/0/0.1
                       [Static/851969] 00:00:09
                       > via sp-3/1/0.1

   root@host# run show route 3040::/16
   inet6.0: 20 destinations, 33 routes (20 active, 0 holddown, 0 hidden)
   + = Active Route, - = Last Active, * = Both

   3040::/16          *[Static/5] 00:00:15
                         via sp-3/0/0.2
                       > via sp-3/1/0.2
                       [Static/786434] 00:23:08
                       > via sp-3/0/0.2
                       [Static/851970] 00:00:14
                       > via sp-3/1/0.2
   ```

BEST PRACTICE: The spd-installed routes have higher metric values (hence a low preference) and the metrics are different. If the metrics are different and ECMP is not enabled, even though multiple routes exist for the same destination, only one of the routes is picked up all the time (based on the metric). For ECMP you must configure equal-cost routes, and hence a manual configuration of routes is needed as shown above.

2. Configure equal-cost multipath (ECMP) load balancing by configuring the hash key at the `[edit forwarding-optionshash-key]` hierarchy level.

```
user@host#  forwarding-options hash-key
[edit forwarding-options hash-key]
user@host#  set family inet layer-3 destination-address
user@host#  set family inet layer-3 source-address
user@host#  set family inet6 layer-3 destination-address
user@host#  set family inet6 layer-3 source-address
```

3. Verify your configuration by displaying `forwarding-options`.

```
user@host# show forwarding-options
hash-key {
    family inet {  <== IPv4 traffic from CEs uses this
        layer-3 {
            destination-address;
            source-address;
        }
    }
    family inet6 {  <== IPv6 traffic from Internet uses this
        layer-3 {
            destination-address;
            source-address;
        }
    }
}
```

TIP: Both IPv4 and IPv6 hash keys must be configured. The IPv4 hash key is used to distribute the traffic coming from CPE devices to the 6rd branch relay. The IPv6 hash key is used to distribute the traffic coming from the IPv6 Internet to the 6rd domain. Because the hash in the forward and reverse direction is for different families, different flows from the same session can reside on different NPUs. However, 6rd processing is stateless (as far as mapping IPv6 packets to softwires is concerned), so this should not be a problem.

Configuring High Availability for 6rd Using 6rd Anycast

You configure 6rd Anycast by defining two service sets that use the same softwire rule in both service sets, just as you do when you configure load balancing for 6rd. However, you do not configure ECMP, and as a result, the services PIC daemon (spd) installs two routes *each* for the softwire concentrator address and 6rd domain pointing to each service interface. The forwarding plane can select any route based on the priority, which is computed when the spd installs the routes. The priority is computed based on the FPC, PIC, slot numbers, and subunit number used on the sp- interface. *Only one PIC is used* based on the route priority, and that PIC gets all of the 6rd traffic. If the PIC goes down. the route pointing to it is also deleted and the forwarding plane automatically selects the alternate available PIC.

6rd Anycast is completely stateless. The spd installs the route and doesn’t run any state machine for the PIC. Because the routes are pre-installed and service sets are already on the PIC, there is no service delay if a failover occurs.

RELATED DOCUMENTATION

Tunneling Services for IPv4-to-IPv6 Transition Overview | **394**

CHAPTER 23

Transition to IPv6 With Inline Softwires

IN THIS CHAPTER

- Inline 6rd and 6to4 Softwires | 451

Inline 6rd and 6to4 Softwires

IN THIS SECTION

- Inline 6rd and 6to4 Configuration Guidelines | 451
- Configuring Inline 6rd | 452
- Examples: 6rd and 6to4 Configurations | 458

Inline 6rd and 6to4 Configuration Guidelines

Keep the following points in mind when you are configuring and using inline 6rd and 6to4.

- You can configure a maximum of 1024 softwire concentrators on a single line card.
- Reassembly of 6rd IPv4 packet from CE is not added as part of this release.
- 6rd multicast is not supported.
- Any ICMPv4 errors generated in the IPv4 access network (between CPE and border relays) are dropped on the border relay. They are not converted into IPv6 errors and forwarded to IPv6 side.
- 6rd/6to4 Anycast and load balancing can be configured only using next-hop style service-interface configuration, not interface style.
- The si- interface input features are not exercised for packets flowing to the 6rd tunnel.

- Bandwidth for traffic from the 6rd tunnel is limited by the available PFE bandwidth; bandwidth for traffic to the 6rd tunnel is limited by the internal VRF loopback bandwidth. SI-IFD loopback bandwidth configuration under the `[edit chassis]` hierarchy has no impact on the 6rd loopback bandwidth.

- If the packet length is more than Tunnel MTU for downlink packets after encapsulating with an IPv4 header, the packet is dropped as v4 MTU errors. For these packet drops an `ICMPv6 packet too big error` message is sent back to the sender. Typically 6rd Tunnel MTU is configured with a high value so if the packet size is larger than the configured value, fragmentation occurs at the egress interface (towards the IPv4 access network).

Configuring Inline 6rd

IN THIS SECTION

- Configuring the Bandwidth for Inline Services | 453
- Configuring the Interfaces | 453
- Configuring the Softwire Concentrator and Rule | 455
- Configuring the Service Set | 457
- Configuring the Routing Instance | 457

Junos OS supports inline 6rd on all Modular Port Concentrator (MPC) line cards on MX Series routers. This saves customers the cost of using MS-DPCs for the required tunneling, encapsulation, and decapsulation processes. Anycast is supported for 6to4 (next-hop service interfaces only). Hairpinning is also supported for traffic between 6rd domains.

Junos OS supports inline 6rd on the following MPCs:

- MPC5 and MPC6—Support starting in Junos OS Release 15.1R3.

- MPC7, MPC8, and MPC9—Support starting in Junos OS Release 17.2R1.

- MPC10E-15C-MRATE and MPC10E-10C-MRATE—Support starting in Junos OS Release 20.3R1.

- MX2K-MPC11E—Support starting in Junos OS Release 20.3R1.

To implement the inline functionality, you configure service interfaces on the MPC as inline services interfaces (si-) rather than as multiServices (ms-) interfaces.

Configuring the Bandwidth for Inline Services

You must provide bandwidth configuration for inline services on the modular port concentrator (MPC) used for inline 6rd processing.

To configure bandwidth:

- Specify the MPC and logical interface, and the desired bandwidth, 1g or 10g.

  ```
  user@host# set chassis fpc mpc-number pic logical-interface-number inline-services bandwidth
  bandwidth
  ```

  For example:

  ```
  user@host# set chassis fpc 0 pic 0 inline-services bandwidth 10g
  ```

Configuring the Interfaces

Configure the si- interfaces for 6rd control and data. 6rd services must be configured on port 0.

To configure the si- interfaces:

1. Configure the 6rd services on port 0 and include units for IPv4 and IPv6.

   ```
   user@host# set interfaces si-mpc-number/logical-interface-number/0 unit 0 family inet
   user@host# set interfaces si-mpc-number/logical-interface-number/0 unit 0 family inet6
   ```

   For example:

   ```
   user@host# set interfaces si-0/0/0 unit 0 family inet
   user@host# set interfaces si-0/0/0 unit 0 family inet6
   ```

2. Configure the media interfaces for the inside service domain.

   ```
   user@host# set interfaces si-mpc-number/logical-interface-number/0 unit unit-number family
   inet
   user@host# set interfaces si-mpc-number/logical-interface-number/0 unit unit-number family
   inet6
   user@host# set interfaces si-0/0/0 unit unit-number service-domain inside
   ```

For example:

```
user@host# set interfaces si-0/0/0 unit 1 family inet
user@host# set interfaces si-0/0/0 unit 1 family inet6
user@host# set interfaces si-0/0/0 unit 1 service-domain inside
```

3. Configure the media interfaces for the outside service domain.

```
user@host# set interfaces si-mpc-number/logical-interface-number/0 unit unit-number family inet
user@host# set interfaces si-mpc-number/logical-interface-number/0 unit unit-number family inet6
user@host# set interfaces si-mpc-number/logical-interface-number/0 unit unit-number service-domain outside
```

For example:

```
user@host# set interfaces si-0/0/0 unit 2 family inet
user@host# set interfaces si-0/0/0 unit 2 family inet family inet6
user@host# set interfaces si-mpc-number/logical-interface-number/0 unit unit-number service-domain outside
```

4. Configure the IPv4-facing interface for use with an interface-style or next-hop service set.

   - To configure for use with an interface-style service set, configure input and output service and specify the service set.

   ```
   user@host# set interfaces ge-mpc-number/logical-interface-number/port unit unit-number family inet service input service-set service-set-name
   user@host# set interfaces ge-mpc-number/logical-interface-number/port unit unit-number family inet service output service-set service-set-name
   user@host# set interfaces ge-mpc-number/logical-interface-number/port unit unit-number family inet address ip-address
   ```

   For example:

   ```
   user@host# set interfaces ge-0/2/7 unit 0 family inet service input service-set vrf-intf-service-set
   user@host# set interfaces ge-0/2/7 unit 0 family inet service output service-set vrf-intf-
   ```

```
service-set
user@host# set interfaces ge-0/2/7 unit 0 family inet address 10.10.10.1/16
```

- To configure for use with a next-hop style service set, omit the `service input` and `service output` references.

  ```
  user@host# set interfaces ge-mpc-number/logical-interface-number/port unit unit-number
  family inet
  user@host# set interfaces ge-mpc-number/logical-interface-number/port unit unit-number
  family inet address ip-address
  ```

  For example:

  ```
  user@host# set interfaces ge-0/2/7 unit 0 family inet
  user@host# set interfaces ge-0/2/7 unit 0 family inet address 10.10.10.1/16
  ```

5. Configure the IPv6 facing interface.

   ```
   user@host# set interfaces ge-mpc-number/logical-interface-number/port unit unit-number family
   inet6 address ipv6-address
   ```

   For example:

   ```
   user@host# set interfaces ge-0/2/8 unit 0 family inet6 address 2001:db8:1:1::1/16
   ```

Configuring the Softwire Concentrator and Rule

Define the softwire concentrator and rule used for encapsulation and decapsulation of IPv6 over IPv4 packets for CE.

To define the softwire concentrator:

1. Specify a 6rd softwire concentrator and its address.

   ```
   user@host# set services softwire softwire-concentrator v6rd concentrator-name softwire-
   address ip-address
   ```

For example:

```
user@host# set services softwire softwire-concentrator v6rd swire01-rd1 softwire-address
10.30.30.1
```

2. Configure the IPv4 address prefix for the customer edge network and the IPv6 address prefix for the 6rd domain.

```
user@host# set services softwire softwire-concentrator v6rd concentrator-name ipv4-prefix
ipv4-prefix v6rd-prefix v6rd-prefix
```

For example:

```
user@host# set services softwire softwire-concentrator v6rd swire01-rd1 ipv4-prefix
10.10.0.0/16 v6rd-prefix 2001:db8:3040::0/48
```

3. Configure the size, in bytes, of the maximum transmission unit `mtu-ipv4` for IPv6 packets encapsulated in IPv4. Compute this as the maximum expected IPv4 packet size plus 20.

```
user@host# set services softwire softwire-concentrator v6rd concentrator-name set mtu-ipv4
number-of-bytes
```

For example:

```
user@host# set services softwire softwire-concentrator v6rd swire01-rd1 set mtu-ipv4 9192
```

To configure the softwire rule:

- Specify the softwire rule, specifying the direction of traffic to be tunneled and the 6rd softwire concentrator to be used.

```
user@host# set services softwire rule softwire-rule-name match-direction match-direction term
rule-term-number then v6rd concentrator-name
```

For example:

```
user@host# set services softwire rule swire01-r1 match-direction input term t1 then v6rd
swire01-rd1
```

Configuring the Service Set

To configure an interface style or next-hop service set for 6rd processing:

- Specify an interface style service set.

  ```
  user@host# set services service-set service-set-name softwire-rules softwire-rule-name service-interface interface-name
  ```

  For example:

  ```
  user@host# set services service-set vrf-intf-service-set softwire-rules swire01-r1 service-interface si-0/0/0.0
  ```

  or

- Configure a next-hop service set.

  ```
  user@host# set services service-set service-set-name softwire-rules softwire-rule-name
  user@host# set services service-set service-set-name next-hop-service inside-service-interface inside-interface outside-service-interface outside-interface
  ```

  ```
  user@host# set services service-set vrf-nh-service-set softwire-rules swire01-r1
  user@host# set services service-set vrf-nh-service-set next-hop-service inside-service-interface si-0/0/0.1 outside-service-interface si-0/0/0.2
  ```

Configuring the Routing Instance

To configure the routing instance:

1. Specify the routing instance and each interface it serves.

   ```
   user@host# set routing-instance routing-instance-name instance-type vrf interface interface-name
   ```

For example:

```
user@host# set routing-instance v6rd-vrf instance-type vrf interface si-0/0/0.1
user@host# set routing-instance v6rd-vrf instance-type vrf interface interface ge-0/2/7.0
```

2. Specify the route distinguisher and vrf-target.

```
user@host# set routing-instance v6rd-vrf route-distinguisher 10.1.1.1:1
user@host# set routing-instance v6rd-vrf vrf-target target:100:100
```

RELATED DOCUMENTATION

Configuring a 6rd Softwire Concentrator | 432

Examples: 6rd and 6to4 Configurations

IN THIS SECTION

- Example: 6rd with Interface-Style Service Set Configuration | 458
- Example: 6rd with Next-Hop-Style Service Set Configuration | 460
- Example: 6rd Anycast Configuration | 462
- Example: Hairpinning Between 6rd Domains Configuration | 465
- Example: 6to4 Configuration | 467

NOTE: The 6rd and 6to4 features are supported on Multiservices 100, 400, and 500 PICs on M Series routers, and on MX Series routers equipped with Multiservices DPCs. MX Series routers with MS-MPCs or MS-MICs support inline 6rd and inline 6to4 features.

Example: 6rd with Interface-Style Service Set Configuration

```
chassis {
    fpc 0 {
        pic 0 {
            inline-services {
                bandwidth 10g;
            }
        }
    }
}
services {
    service-set vrf-intf-service-set {
        softwire-rules swire01-r1;
        interface-service {
            service-interface si-0/0/0.0;
        }
    }
    softwire {
        softwire-concentrator {
            v6rd swire01-rd1 {
                softwire-address 10.30.30.1;
                ipv4-prefix 10.10.0.0/16;
                v6rd-prefix 2001:db8::/32;
                mtu-v4 9192;
            }
        }
        rule swire01-r1 {
            match-direction input;
            term t1 {
                then {
                    v6rd swire01-rd1;
                }
            }
        }
    }
}
interfaces {
    si-0/0/0 {
        unit 1 {
            family inet;
            family inet6;
            service-domain inside;
        }
        unit 2 {
            family inet;
            family inet6;
            service-domain outside;
        }
    }

    ge-0/2/7 {
        unit 0 {
            family inet {
                address 10.10.10.1/16;
            }
        }
    }
    ge-0/2/8 {
        unit 0 {
            family inet6 {
                address 2001:db8:3abc::1/64;
            }
        }
    }
}
routing-instances {
    v6rd-vrf {
        instance-type vrf;
        interface si-0/0/0.1;
        interface ge-0/2/7.0;
        route-distinguisher 10.1.1.1:1;
        vrf-target target:100:100;
    }
}
```

Example: 6rd with Next-Hop-Style Service Set Configuration

```
chassis {
    fpc 0 {
        pic 0 {
            inline-services {
                bandwidth 10g;
            }
        }
    }
}
services {
    service-set vrf-nh-service-set {
        softwire-rules swire01-r1;
        next-hop-service {
            inside-service-interface si-0/0/0.1;
            outside-service-interface si-0/0/0.2;
        }
    }
    softwire {
        softwire-concentrator {
            v6rd swire01-rd1 {
                softwire-address 10.30.30.1;
                ipv4-prefix 10.10.0.0/16;
                v6rd-prefix 2001:db8:3040::0/48;
                mtu-v4 9192;
            }
        }
        rule swire01-r1 {
            match-direction input;
            term t1 {
                then {
                    v6rd swire01-rd1;
                }
            }
        }
    }
}
interfaces {
    si-0/0/0 {
        unit 1 {
            family inet;
            family inet6;
            service-domain inside;
        }
        unit 2 {
            family inet;
            family inet6;
            service-domain outside;
        }
    }

    ge-0/2/7 {
        unit 0 {
            family inet {
                address 10.10.10.1/16;
            }
        }
    }
    ge-0/2/8 {
        unit 0 {
            family inet6 {
                address 2001:db8:3abc::1/64;
            }
        }
    }
}
routing-instances {
    v6rd-vrf {
        instance-type vrf;
        interface si-0/0/0.1;
        interface ge-0/2/7.0;
        route-distinguisher 10.1.1.1:1;
        vrf-target target:100:100;
    }
}
```

Example: 6rd Anycast Configuration

```
chassis {
    fpc 0 {
        pic 0 {
            inline-services {
                bandwidth 10g;
            }
        }
        pic 2 {
            inline-services {
                bandwidth 1g;
            }
        }
    }
}
services {
    service-set anycast-nh-set1 {
        softwire-rules swire01-r1;
        next-hop-service {
            inside-service-interface si-0/0/0.1;
            outside-service-interface si-0/0/0.2;
        }
    }
    service-set anycast-nh-set2 {
        softwire-rules swire01-r1;
        next-hop-service {
            inside-service-interface si-0/2/0.1;
            outside-service-interface si-0/2/0.2;
        }
    }
    softwire {
        softwire-concentrator {
            v6rd swire01-rd1 {
                softwire-address 10.30.30.1;
                ipv4-prefix 10.10.0.0/16;
                v6rd-prefix 2001:db8:3040::0/48;
                mtu-v4 9192;
            }
        }
        rule swire01-r1 {
            match-direction input;
            term t1 {
                then {
                    v6rd swire01-rd1;
                }
            }
        }
    }
}
interfaces {
    si-0/0/0 {
        unit 0 {
            family inet;
            family inet6;
        }
        unit 1 {
            family inet;
            family inet6;
            service-domain inside;
        }
        unit 2 {
            family inet;
            family inet6;
            service-domain outside;
        }
    }
    si-0/2/0 {
        unit 0 {
            family inet;
            family inet6;
        }
        unit 1 {
            family inet;
            family inet6;
            service-domain inside;
        }
        unit 2 {
            family inet;
            family inet6;
            service-domain outside;
        }
    }
    ge-0/2/7 {
        unit 0 {
            family inet {
                address 10.10.10.1/16;
            }
        }
    }
    ge-0/2/8 {
        unit 0 {
            family inet6 {
                address 2001:db8:3abc::1/64;
            }
        }
    }
}
```

Example: Hairpinning Between 6rd Domains Configuration

This example uses an interface service-set and a next-hop service set as hairpinning domains.

```
chassis {
    fpc 0 {
        pic 0 {
            inline-services {
                bandwidth 10g;
            }
        }
    }
}
services {
    service-set hairpin-intf-service-set {
        softwire-rules swire01-r1;
        interface-service {
            service-interface si-0/0/0.0;
        }
    }
    service-set hairpin-nh-service-set {
        softwire-rules swire01-r2;
        next-hop-service {
            inside-service-interface si-0/0/0.1;
            outside-service-interface si-0/0/0.2;
        }
    }
    softwire {
        softwire-concentrator {
            v6rd swire01-rd1 {
                softwire-address 30.30.30.1;
                ipv4-prefix 10.10.0.0/16;
                v6rd-prefix 2001:db8:3040::0/48;
                mtu-v4 9192;
            }
            v6rd swire01-rd2 {
                softwire-address 10.60.60.1;
                ipv4-prefix 10.40.40.0/24;
                v6rd-prefix 2001:db8:3050::0/48;
                mtu-v4 9192;
            }
        }
        rule swire01-r1 {
            match-direction input;
            term t1 {
                then {
                    v6rd swire01-rd1;
                }
            }
        }
        rule swire01-r2 {
            match-direction input;
            term t1 {
                then {
                    v6rd swire01-rd2;
                }
            }
        }
    }
}
interfaces {
    si-0/0/0 {
        unit 0 {
            family inet;
            family inet6;
        }
        unit 1 {
            family inet;
            family inet6;
            service-domain inside;
        }
        unit 2 {
            family inet;
            family inet6;
            service-domain outside;
        }
    }
    ge-0/2/7 {
        unit 0 {
            family inet {
                service {
                    input {
                        service-set hairpin-intf-service-set;
                    }
                    output {
                        service-set hairpin-intf-service-set;
                    }
                }
                address 10.10.10.1/16;
            }
        }
    }
    ge-0/2/8 {
        unit 0 {
            family inet {
                address 10.40.40.1/24;
            }
        }
    }
}
```

Example: 6to4 Configuration

```
chassis {
    fpc 0 {
        pic 0 {
            inline-services {
                bandwidth 10g;
            }
        }
    }
}
services {
    service-set 6to4-intf-service-set {
        softwire-rules shenick01-r1;
        interface-service {
            service-interface si-0/0/0.0;
        }
    }
    interfaces {
        si-0/0/0 {
            unit 0 {
                family inet;
                family inet6;
            }
            unit 1 {
                family inet;
                family inet6;
                service-domain inside;
            }
            unit 2 {
                family inet;
                family inet6;
                service-domain outside;
            }
        }
        ge-0/2/7 {
            unit 0 {
                family inet {
                    service {
                        input {
                            service-set 6to4-intf-service-set;
                        }
                        output {
                            service-set 6to4-intf-service-set;
                        }
                    }
                    address 10.10.10.1/16;
                }
            }
        }
        ge-0/2/8 {
            unit 0 {
                family inet6 {
                    address 2001:db8:3abc::1/64;
                }
            }
        }
    }
```

SEE ALSO

Configuring a 6to4 Provider-Managed Tunnel

CHAPTER 24

Monitoring and Troubleshooting Softwires

IN THIS CHAPTER

- Monitoring and Troubleshooting Softwires | 469

Monitoring and Troubleshooting Softwires

IN THIS SECTION

- Ping and Traceroute for DS-Lite | 469
- Monitoring Softwire Statistics | 470
- Monitoring CGN, Stateful Firewall, and Softwire Flows | 472

Ping and Traceroute for DS-Lite

With Junos OS Release 11.4, you can use the **ping** and `traceroute` commands to determine the status of the DS-Lite softwire tunnels:

- IPv6 ping—The softwire address endpoint on the DS-Lite softwire terminator (AFTR) is usually configured only at the `[edit services softwire]` hierarchy level; it need not be hosted on any interface. Previous releases of the Junos OS software did not provide replies to pings to the IPv6 softwire address when the AFTR was not configured on a specific interface or loopback. An IPv6 ping enables the softwire initiator (B4) to verify the softwire address of the AFTR before creating a tunnel.

- IPv4 ping—A special IPv4 address, 192.0.0.1, is reserved for the AFTR. Previous releases of the Junos OS did not respond to any pings sent to this address. A B4 and other IPv4 nodes can now ping to this address to determine whether the DS-Lite tunnel is working.

- Traceroute—The AFTR now generates and forwards traceroute packets over the DS-Lite tunnel.

NOTE: No additional CLI configuration is necessary to use the new functionality.

Monitoring Softwire Statistics

IN THIS SECTION

- Purpose | 470
- Action | 470

Purpose

You can review softwire global statistics by using the **show services softwire** or `show services softwire statistics` command.

Action

```
user@host# show services softwire
Interface: sp-0/0/0, Service set: sset
Softwire Direction Flow count
2001:0:0:1::1 -> 1001::1 I 3
```

```
user@host# show services softwire statistics
DS-Lite Statistics:
Service PIC Name: :sp-0/0/0
Statistics
----------
Softwires Created :2
Softwires Deleted :1
Softwires Flows Created :2
Softwires Flows Deleted :1
Slow Path Packets Processed :2
Fast Path Packets Processed :274240
Fast Path Packets Encapsulated :583337
Rule Match Failed :0
Rule Match Succeeded :2
```

```
IPv6 Packets Fragmented :0
Transient Errors
----------------
Flow Creation Failed - Retry :0
Slow Path Failed - Retry :0
Errors
------
Softwire Creation Failed :0
Flow Creation Failed :0
Slow Path Failed :0
Packet not IPv4-in-IPv6 :0
IPv6 Fragmentation Error :0
Slow Path Failed - IPv6 Next Header Offset :0
Decapsulated Packet not IPv4 :0
Fast Path Failed - IPv6 Next Header Offset :0
No Softwire ID :0
No Flow Extension :0
Flow Limit Exceeded :0
6rd Statistics:
Service PIC Name :sp-0/0/0
Statistics
----------
Softwires Created :0
Softwires Deleted :0
Softwires Flows Created :0
Softwires Flows Deleted :0
Slow Path Packets Processed :0
Fast Path Packets Processed :0
Fast Path Packets Encapsulated :0
Rule Match Failed :0
Rule Match Succeeded :0
Transient Errors
----------------
Flow Creation Failed - Retry :0
Slow Path Failed - Retry :0
Errors
------
Softwire Creation Failed :0
Flow Creation Failed :0
Slow Path Failed :0
Packet not IPv6-in-IPv4 :0
Slow Path Failed - IPv6 Next Header Offset :0
Decapsulated Packet not IPv6 :0
```

```
Encapsulation Failed - No packet memory :0
No Softwire ID :0
No Flow Extension :0
ICMPv4 Dropped Packets :0
```

Monitoring CGN, Stateful Firewall, and Softwire Flows

IN THIS SECTION

- Purpose | **472**
- Action | **472**

Purpose

Use the following commands to check the creation of the softwires, pre-NAT flows, and post-NAT flows. Output can be filtered using more specific fields such as AFTR or B4 address or both for DS-Lite, and softwire-concentrator or softwire-initiator or both for 6rd.

- show services stateful-firewall flows
- show services softwire flows

Action

```
user@host# show services stateful-firewall flows
Interface: sp-0/1/0, Service set: dslite-svc-set2
Flow                                                State    Dir       Frm count
TCP      200.200.200.2:80    ->     44.44.44.1:1025  Forward  O          219942
    NAT dest        44.44.44.1:1025    ->       20.20.1.4:1025
    Softwire           2001::2         ->         1001::1
TCP          20.20.1.2:1025  ->  200.200.200.2:80    Forward  I          110244
    NAT source       20.20.1.2:1025    ->      44.44.44.1:1024
    Softwire           2001::2         ->         1001::1
TCP      200.200.200.2:80    ->     44.44.44.1:1024  Forward  O          219140
    NAT dest        44.44.44.1:1024    ->       20.20.1.2:1025
    Softwire           2001::2         ->         1001::1
DS-LITE         2001::2      ->        1001::1       Forward  I          988729
TCP      200.200.200.2:80    ->     44.44.44.1:1026  Forward  O          218906
    NAT dest        44.44.44.1:1026    ->       20.20.1.3:1025
    Softwire           2001::2         ->         1001::1
TCP          20.20.1.3:1025  ->  200.200.200.2:80    Forward  I          110303
    NAT source       20.20.1.3:1025    ->      44.44.44.1:1026
    Softwire           2001::2         ->         1001::1
TCP          20.20.1.4:1025  ->  200.200.200.2:80    Forward  I          110944
    NAT source       20.20.1.4:1025    ->      44.44.44.1:1025
    Softwire           2001::2         ->         1001::1
```

SEE ALSO

Tunneling Services for IPv4-to-IPv6 Transition Overview | 394

5

PART

ALGs

ALGs | 475

CHAPTER 25

ALGs

IN THIS CHAPTER

- ALG Overview | 475
- ALG Applications | 507

ALG Overview

IN THIS SECTION

- ALG Descriptions | 475
- ICMP, Ping, and Traceroute ALGs for MS-MICs and MS-MPCs | 506

ALG Descriptions

IN THIS SECTION

- Supported ALGs | 476
- ALG Support Details | 478
- Juniper Networks Defaults | 490
- Examples: Referencing the Preset Statement from the Junos OS Default Group | 504

This topic describes the Application Layer Gateways (ALGs) supported by Junos OS. ALG support includes managing pinholes and parent-child relationships for the supported ALGs.

Supported ALGs

Table 15 on page 476 lists ALGs supported by Junos OS. For information about which ALGs are supported on MS-DPCs, MS-MPCs, MS-MICs, see "ALGs Available for Junos OS Address Aware NAT" on page 71 .

Table 15: ALGs Supported by Junos OS

ALGs Supported	v4 - v4	v6 - v4	v6 - v6	DS-Lite (Support for ALGs with DS-lite on MS-MPC and MS-MIC starts in Junos OS Release 18.1R1)
Basic TCP ALG	Yes	Yes	Yes	Yes
Basic UPD ALG	Yes	Yes	Yes	Yes
BOOTP	Yes	No	No	No
DCE RPC Services	Yes	No	No	No
DNS	Yes	Yes	No	Yes
FTP	Yes	Yes (Starting in Junos OS Release 14.1R1)	No	Yes
Gatekeeper RAS	Yes (Starting in Junos OS Release 17.1R1)	Yes (Starting in Junos OS Release 17.2R1)	No	No
H323	Yes	Yes (Starting in Junos OS Release 17.2R1)	No	No
ICMP	Yes	Yes	Yes	Yes

Table 15: ALGs Supported by Junos OS *(Continued)*

ALGs Supported	v4 - v4	v6 - v4	v6 - v6	DS-Lite (Support for ALGs with DS-lite on MS-MPC and MS-MIC starts in Junos OS Release 18.1R1)
IKE ALG (Starting in Junos OS Release 14.2R7, 15.1R5, 16.1R2, and 17.1R1)	Yes	Yes	No	No
IIOP	Yes	No	No	No
IP	Yes	No	No	No
NETBIOS	Yes	No	No	No
NETSHOW	Yes	No	No	No
PPTP	Yes	Yes (Starting in Junos OS Release 14.1R1)	No	Yes
REALAUDIO	Yes	No	No	No
Sun RPC and RPC Port Map Services	Yes	No	No	No
RTSP	Yes	Yes (Starting in Junos OS Release 14.1R1)	No	Yes

Table 15: ALGs Supported by Junos OS *(Continued)*

ALGs Supported	v4 - v4	v6 - v4	v6 - v6	DS-Lite (Support for ALGs with DS-lite on MS-MPC and MS-MIC starts in Junos OS Release 18.1R1)
SIP	Yes	Yes (Starting in Junos OS Release 14.1R1)	No	SIP supported for DS-Lite on MS-MPC and MS-MIC starting in Junos OS Release 18.2R1.
SNMP	Yes	No	No	No
SQLNET	Yes	No	No	No
TFTP	Yes	Yes (Starting in Junos OS Release 14.1R1)	No	Yes
Traceroute	Yes	Yes	No	Yes
Unix Remote Shell Service	Yes	No	No	No
WINFrame	Yes	No	No	No

ALG Support Details

This section includes details about the ALGs. It includes the following:

Basic TCP ALG

This ALG performs basic sanity checking on TCP packets. If it finds errors, it generates the following anomaly events and system log messages:

- TCP source or destination port zero
- TCP header length check failed
- TCP sequence number zero and no flags are set
- TCP sequence number zero and FIN/PSH/RST flags are set
- TCP FIN/RST or SYN(URG|FIN|RST) flags are set

The TCP ALG performs the following steps:

1. When the router receives a SYN packet, the ALG creates TCP forward and reverse flows and groups them in a *conversation*. It tracks the TCP three-way handshake.
2. The SYN-defense mechanism tracks the TCP connection establishment state. It expects the TCP session to be established within a small time interval (currently 4 seconds). If the TCP three-way handshake is not established in that period, the session is terminated.
3. A keepalive mechanism detects TCP sessions with nonresponsive endpoints.
4. ICMP errors are allowed only when a flow matches the selector information specified in the ICMP data.

Basic UDP ALG

This ALG performs basic sanity checking on UDP headers. If it finds errors, it generates the following anomaly events and system log messages:

- UDP source or destination port 0
- UDP header length check failed

The UDP ALG performs the following steps:

1. When it receives the first packet, the ALG creates bidirectional flows to accept forward and reverse UDP session traffic.
2. If the session is idle for more than the maximum allowed idle time (the default is 30 seconds), the flows are deleted.
3. ICMP errors are allowed only when a flow matches the selector information specified in the ICMP data.

BOOTP

The Bootstrap Protocol (BOOTP) client retrieves its networking information from a server across the network. It sends out a general broadcast message to request the information, which is returned by the BOOTP server. For the protocol specification, see `ftp://ftp.isi.edu/in-notes/rfc951.txt`.

Stateful firewall support requires that you configure the BOOTP ALG on UDP server port 67 and client port 68. If the client sends a broadcast message, you should configure the broadcast address in the `from` statement of the service rule. Network Address Translation (NAT) is not performed on the BOOTP traffic, even if the NAT rule matches the traffic. If the BOOTP relay feature is activated on the router, the remote BOOTP server is assumed to assign addresses for clients masked by NAT translation.

DCE RPC Services

Distributed Computing Environment (DCE) Remote Procedure Call (RPC) services are mainly used by Microsoft applications. The ALG uses well-known TCP port 135 for port mapping services, and uses the universal unique identifier (UUID) instead of the program number to identify protocols. The main application-based DCE RPC is the Microsoft Exchange Protocol.

Support for stateful firewall and NAT services requires that you configure the DCE RPC portmap ALG on TCP port 135. The DCE RPC ALG uses the TCP protocol with application-specific UUIDs.

DNS

The Domain Name System (DNS), which typically runs on port 53, handles the data associated with locating and translating domain names into IP addresses. The MX Series DNS ALG monitors the DNS query and reply packets, and supports UDP and TCP DNS traffic independently. The DNS ALG does not support payload translations for NAT, but an operator can use it to efficiently remove the NAT or stateful firewall DNS sessions from memory after the DNS server sends its response. The DNS ALG closes the session only when a reply is received or an idle timeout is reached.

There might be issues with TCP DNS traffic when the TCP-DNS-ALG is used if the DNS traffic is not just the standard request and reply type. For example, the TCP-DNS-ALG might break DNS server-to-server communication that uses TCP, such as DNS Replication or Zone transfers. This type of traffic might get dropped by the NAT or stateful firewall plugins because the TCP-DNS-ALG closes the session after the TCP handshake is complete and after each server has sent one packet to the other. In these instances do not use the TCP-DNS-ALG.

NOTE:
The TCP-DNS-ALG is not supported on the MS-DPC service cards.

FTP

FTP is the File Transfer Protocol, specified in RFC 959. In addition to the main control connection, data connections are also made for any data transfer between the client and the server; and the host, port, and direction are negotiated through the control channel.

For non-passive-mode FTP, Junos OS stateful firewall service scans the client-to-server application data for the PORT command, which provides the IP address and port number to which the server connects. For passive-mode FTP, Junos OS stateful firewall service scans the client-to-server application data for the PASV command and then scans the server-to-client responses for the 227 response, which contains the IP address and port number to which the client connects.

There is an additional complication: FTP represents these addresses and port numbers in ASCII. As a result, when addresses and ports are rewritten, the TCP sequence number might be changed, and thereafter the NAT service needs to maintain this delta in SEQ and ACK numbers by performing sequence NAT on all subsequent packets.

Support for stateful firewall and NAT services requires that you configure the FTP ALG on TCP port 21 to enable the FTP control protocol. The ALG performs the following tasks:

- Automatically allocates data ports and firewall permissions for dynamic data connection

- Creates flows for the dynamically negotiated data connection

- Monitors the control connection in both active and passive modes

- Rewrites the control packets with the appropriate NAT address and port information

On MS-MPCs, MS-MICs, for passive FTP to work properly without FTP application layer gateway (ALG) enabled (by not specifying the `application junos-ftp` statement at the `[edit services stateful-firewall rule` *`rule-name`* `term` *`term-name`* `from]` and the `[edit services nat rule` *`rule-name`* `term` *`term-name`* `from]` hierarchy levels), you must enable the address pooling paired (APP) functionality enabled (by including the `address-pooling` statement at the `[edit services nat rule` *`rule-name`* `term` *`term-name`* `then translated]` hierarchy level). Such a configuration causes the data and control FTP sessions to receive the same NAT address.

Gatekeeper RAS

Starting in Junos OS Release 17.1R1, the gatekeeper registration, administration, and status (RAS) ALG allows full support of gatekeeper mode for H.323 calls. An endpoint registers to a gatekeeper and asks for its management. Before making a call, an endpoint asks its gatekeeper for permission to place the call. In both registration and admission phases, the RAS channel is used. Use the gatekeeper RAS ALG and the H323 ALG in IPv4 and IPv6 stateful-firewall rules or NAPT-44 rules. Starting in Junos OS Release 17.2R1, NAT-64 rules are also supported. The Junos default application set `junos-h323-suite` includes the H323 ALG and the gatekeeper RAS ALG.

H323

H323 is a suite of ITU protocols for audio and video conferencing and collaboration applications. H323 consists of H.225 call signaling protocols and H.245 control protocol for media communication. During H.225 negotiation, the endpoints create a call by exchanging call signaling messages on the control channel and negotiate a new control channel for H.245. A new control connection is created for H.245 messages. Messages are exchanged on the H.245 control channel to open media channels.

Stateful firewall monitors the H.225 control channel to open the H.245 control channel. After the H.245 channel is created, stateful firewall also monitors this channel for media channel information and allows the media traffic through the firewall.

H323 ALG supports static destination, static and dynamic source NAT by rewriting the appropriate addresses and ports in the H.225 and H.245 messages.

To support gatekeeper mode for H.323 calls, use the H323 ALG and the gatekeeper RAS ALG in IPv4 and IPv6 stateful-firewall rules or NAPT-44 rules. Starting in Junos OS Release 17.2R1, NAT-64 rules are also supported. The Junos default application set `junos-h323-suite` includes the H323 ALG and the gatekeeper RAS ALG.

ICMP

The Internet Control Message Protocol (ICMP) is defined in RFC 792. The Junos OS stateful firewall service allows ICMP messages to be filtered by specific type or specific type code value. ICMP error packets that lack a specifically configured type and code are matched against any existing flow in the opposite direction to check for the legitimacy of the error packet. ICMP error packets that pass the filter matching are subject to NAT translation.

The ICMP ALG always tracks ping traffic statefully using the ICMP sequence number. Each echo reply is forwarded only if there is an echo request with the corresponding sequence number. For any ping flow, only 20 echo requests can be forwarded without receiving an echo reply. When you configure dynamic NAT, the PING packet identifier is translated to allow additional hosts in the NAT pool to use the same identifier.

Support for stateful firewall and NAT services requires that you configure the ICMP ALG if the protocol is needed. You can configure the ICMP type and code for additional filtering.

IIOP

The Oracle Application Server Name Server Internet Inter-ORB Protocol (IIOP). This ALG is used in Common Object Request Broker Architecture (CORBA) based on distributed computing. Even though CORBA and IIOP are Object Management Group (OMG) standards, no fixed port is assigned for IIOP. Each vendor implementing CORBA chooses a port. Java Virtual machine uses port 1975 by default, while ORBIX uses port 3075 as a default.

Stateful firewall and NAT require ALG IIOP be configured for TCP port 1975 for Java VM IIOP, and 3075 for CORBA applications ORBIX, a CORBA framework from Iona Technologies.

IKE ALG

Before Junos OS Release 17.4R1, Network Address Translation-Traversal (NAT-T) is not supported for the Junos VPN Site Secure suite of IPsec features on the MX Series routers. Starting in Junos OS Release 14.2R7, 15.1R5, 16.1R2, and 17.1R1, the IKE ALG enables the passing of IKEv1 and IPsec packets through NAPT-44 and NAT64 rules between IPsec peers that are not NAT-T compliant. This ALG supports only ESP tunnel mode.

Use this ALG in NAT rules and specify the UDP protocol and port 500.

This ALG performs the following:

- Tracks IKEv1 connection-initiation requests to determine whether NAT processing is required.
- Performs NAT translation on outgoing and incoming IKEv1 requests and creates IKE sessions.
- Identifies IPsec packets related to the established IKE session and establishes security association between peers.
- Performs NAT translation on IPsec packets.

IP

The IP ALG is used to create unidirectional flows only. In case of TCP traffic, it does not check the 3-way handshake process. This ALG is useful in case of stateful firewall only service sets, where it allows traffic to flow uni-directionally only. When configuring in conjunction with `match-direction input-output` it allows the return traffic to flow through the stateful firewall as well. Typical scenarios are static NAT, destination NAT or scenarios where traffic is expected to traverse the stateful firewall in the presence of asymmetric routing. The Junos IP ALG is not intended for use with NAPT, which causes matching traffic to be discarded through the creation of a drop flow.

NetBIOS

A NetBIOS ALG translates NetBIOS IP addresses and port numbers when NAT is used.

NetBIOS supports the TCP and UDP transport protocols. Support for stateful firewall and NAT services requires that you configure the NetBIOS ALG on UDP port 138 and TCP port 139.

NetShow

The Microsoft protocol ms-streaming is used by NetShow, the Microsoft media server. This protocol supports several transport protocols: TCP, UDP, and HTTP. The client starts a TCP connection on port 1755 and sends the PORT command to the server. The server then starts UDP on that port to the client. Support for stateful firewall and NAT services requires that you configure the NetShow ALG on UDP port 1755.

ONC RPC Services

Open Networks Computing (ONC) RPC services function similarly to DCE RCP services. However, the ONC RPC ALG uses TCP/UDP port 111 for port mapping services, and uses the program number to identify protocols rather than the UUID.

Support for stateful firewall and NAT services requires that you configure the ONC RPC portmap ALG on TCP port 111. The ONC RPC ALG uses the TCP protocol with application-specific program numbers.

PPTP

The Point-to-Point Tunneling Protocol (PPTP) ALG is a TCP-based ALG. PPTP allows the Point-to-Point Protocol (PPP) to be tunneled through an IP network. PPTP defines a client-server architecture, a PPTP Network Server, and a PPTP Access Concentrator. The PPTP ALG requires a control connection and a data tunnel. The control connection uses TCP to establish and disconnect PPP sessions, and runs on port 1723. The data tunnel carries PPP traffic in generic routing encapsulated (GRE) packets that are carried over IP.

RealAudio

Real Networks PNA protocol RealVideo is not a separate service. It is part of the RealPlayer and most likely uses another channel for video. The RealPlayer versions G2, 7, and 8 use PNA and RTSP. For this version to work, the ALG must allow both PNA(7070) and RTSP(554). For the media, the server selects from a range of UDP ports(6970 through 7170), or TCP port 7071, or HTTP. The client can be configured to use a particular port. The RealPlayer versions 4.0 and 5.0 use control channel 7070 media UDP ports 6970 through 7170, or TCP port 7071, or HTTP. RealAudio player version 3.0 uses control channel 7070 media, UDP ports 6770-7170, or TCP port 7071.

Real products use the ports and ranges of ports shown in Table 16 on page 484 .

Table 16: RealAudio Product Port Usage

Real Product	Port Usage
4.0 and 5.0 Servers/ Players	Control channel (bidirectional) on TCP port 7070. Data channel from server to player on TCP port 7070 or UDP port 6970-7170.

Table 16: RealAudio Product Port Usage *(Continued)*

Real Product	Port Usage
4.0 and 5.0 Servers/ Encoders	Control channel (bidirectional) on TCP port 7070. Data channel from encoder or server on TCP port 7070.
G2 Servers/Players	Control channel (bidirectional) on TCP port 80, 554, 7070, or 8080. Data channel from server to player on TCP port 80, 554, 7070, 8080 or UDP port 6970-32,000.
G2 Server/3.1, and 5.x Encoders	Control channel (bidirectional) on TCP port 7070. Data channel from encoder to server on TCP port 7070.
G2 Server/G2 Producer	Control channel (bidirectional) on TCP port 4040. Data channel from encoder to server on TCP port 4040 and UDP port 6970-32,000.
2 Server/G2 Producer (TCP ONLY)	Control channel (bidirectional) on TCP port 4040 Data channel from encoder to server on TCP port 4040. Note: TCP-ONLY option available in version 6.1 or above.

NOTE: RealAudio was the original protocol by RealPlayers. Newer versions of RealPlayer use RTSP. Stateful firewall and NAT require ALG RealAudio to be programmed on TCP port 7070.

Sun RPC and RPC Portmap Services

The Remote Procedure Call (RPC) ALG uses well-known ports TCP 111 and UDP 111 for port mapping, which dynamically assigns and opens ports for RPC services. The RPC Portmap ALG keeps track of port requests and dynamically opens the firewall for these requested ports. The RPC ALG can further restrict the RPC protocol by specifying allowed program numbers.

The ALG includes the RPC services listed in Table 17 on page 486 .

Table 17: Supported RPC Services

Name	Description	Comments
`rpc-mountd`	Network File Server (NFS) mount daemon; for details, see the UNIX man page for `rpc.mountd(8)`.	The base support is RPC v2 and the port mapper service on port 111 (see RFC 1050).
`rpc-nfsprog`	Used as part of NFS. For details, see RFC 1094. See also RFC1813 for NFS v3.	The base support is RPC v2 and the port mapper service on port 111 (see RFC 1050).
`rpc-nisplus`	Network Information Service Plus (NIS+), designed to replace NIS; it is a default naming service for Sun Solaris and is not related to the old NIS. No protocol information is available.	The base support is RPC v2 and the port mapper service on port 111 (see RFC 1050).
`rpc-nlockmgr`	Network lock manager.	The base support is RPC v2 and the port mapper service on port 111 (see RFC 1050). Once the RPC program table is built, `rpc-nlockmgr` service can be allowed or blocked based on RPC program 100021.
`rpc-pcnfsd`	Kernel statistics server. For details, see the UNIX man pages for `rstatd` and `rpc.rstatd`.	The base support is RPC v2 and the port mapper service on port 111 (see RFC 1050). Once the RPC program table is built, `rpc-rstat` service can be allowed or blocked based on RPC program 150001.
`rpc-rwall`	Used to write a message to users; for details, see the UNIX man page for `rpc.rwalld`.	The base support is RPC v2 and the port mapper service on port 111 (see RFC 1050). Once the RPC program table is built, `rpc-rwall` service can be allowed or blocked based on RPC program 150008.
`rpc-ypbind`	NIS binding process. For details, see the UNIX man page for `ypbind`.	The base support is RPC v2 and the port mapper service on port 111 (see RFC 1050). Once the RPC program table is built, `rpc-ypbind` service can be allowed or blocked based on RPC program 100007.

Table 17: Supported RPC Services *(Continued)*

Name	Description	Comments
`rpc-yppasswd`	NIS password server. For details, see the UNIX man page for `yppasswd`.	The base support is RPC v2 and the port mapper service on port 111 (see RFC 1050). Once the RPC program table is built, `rpc-yppasswd` service can be allowed or blocked based on RPC program 100009.
`rpc-ypserv`	NIS server. For details, see the UNIX man page for `ypserv`.	The base support is RPC v2 and the port mapper service on port 111 (see RFC 1050). Once the RPC program table is built, `rpc-ypserv` service can be allowed or blocked based on RPC program 100004.
`rpc-ypupdated`	Network updating tool.	The base support is RPC v2 and the port mapper service on port 111 (see RFC 1050). Once the RPC program table is built, `rpc-ypupdated` service can be allowed or blocked based on RPC program 100028.
`rpc-ypxfrd`	NIS map transfer server. For details, see the UNIX man page for `rpc.ypxfrd`.	The base support is RPC v2 and the port mapper service on port 111 (see RFC 1050). Once the RPC program table is built, `rpc-ypxfrd` service can be allowed or blocked based on RPC program 100069.

Support for stateful firewall and NAT services that use port mapping requires that you configure the RPC portmap ALG on TCP/UDP destination port 111 and the RPC ALG for both TCP and UDP. You can specify one or more `rpc-program-number` values to further restrict allowed RPC protocols.

RTSP

The Real-Time Streaming Protocol (RTSP) controls the delivery of data with real-time properties such as audio and video. The streams controlled by RTSP can use RTP, but it is not required. Media can be transmitted on the same RTSP control stream. This is an HTTP-like text-based protocol, but client and server maintain session information. A session is established using the SETUP message and terminated using the TEARDOWN message. The transport (the media protocol, address, and port numbers) is negotiated in the setup and the setup-response.

Support for stateful firewall and NAT services requires that you configure the RTSP ALG for TCP port 554.

The ALG monitors the control connection, opens flows dynamically for media (RTP/RTSP) streams, and performs NAT address and port rewrites.

SIP

The Session Initiation Protocol (SIP) is an application layer protocol that can establish, maintain, and terminate media sessions. It is a widely used voice over IP (VoIP) signaling protocol. The SIP ALG monitors SIP traffic and dynamically creates and manages pinholes on the signaling and media paths. The ALG only allows packets with the correct permissions. The SIP ALG also performs the following functions:

- Manages parent-child session relationships.
- Enforces security policies.
- Manages pinholes for VoIP traffic.

The SIP ALG supports the following features:

- Stateful firewall
- Static source NAT
- Dynamic address only source NAT
- *Network Address Port Translation* (NAPT)

NOTE: SIP sessions are limited to 12 hours (720 minutes) for NAT processing on the MS-MIC and MS-MPC interface cards. SIP sessions on the MS-DPC have no time limit.

SNMP

SNMP is a communication protocol for managing TCP/IP networks, including both individual network devices and aggregated devices. The protocol is defined by RFC 1157. SNMP runs on top of UDP.

The Junos OS stateful firewall service implements the SNMP ALG to inspect the SNMP type. SNMP does not enforce stateful flow. Each SNMP type needs to be specifically enabled. Full SNMP support of stateful firewall services requires that you configure the SNMP ALG on UDP port 161. This enables the SNMP *get* and `get-next` commands, as well as their response traffic in the reverse direction: UDP port 161 enables the SNMP `get-response` command. If SNMP traps are permitted, you can configure them on UDP port 162, enabling the SNMP `trap` command.

SQLNet

The SQLNet protocol is used by Oracle SQL servers to execute SQL commands from clients, including load balancing and application-specific services.

Support of stateful firewall and NAT services requires that you configure the SQLNet ALG for TCP port 1521.

The ALG monitors the control packets, opens flows dynamically for data traffic, and performs NAT address and port rewrites.

TFTP

The Trivial File Transfer Protocol (TFTP) is specified in RFC 1350. The initial TFTP requests are sent to UDP destination port 69. Additional flows can be created to **get** or **put** individual files. Support of stateful firewall and NAT services requires that you configure the TFTP ALG for UDP destination port 69.

Traceroute

Traceroute is a tool for displaying the route that packets take to a network host. It uses the IP time-to-live (TTL) field to trigger ICMP time-exceeded messages from routers or gateways. It sends UDP datagrams to destination ports that are believed to be not in use; destination ports are numbered using the formula: + *n*hops – 1. The default base port is 33434. To support traceroute through the firewall, two types of traffic must be passed through:

1. UDP probe packets (UDP destination port > 33000, IP TTL < 30)

2. ICMP response packets (ICMP type time-exceeded)

When NAT is applied, the IP address and port within the ICMP error packet also must be changed.

Support of stateful firewall and NAT services requires you to configure the Traceroute ALG for UDP destination port 33434 to 33450. In addition, you can configure the TTL threshold to prevent UDP flood attacks with large TTL values.

UNIX Remote-Shell Services

Three protocols form the basis for UNIX remote-shell services:

- Exec—Remote command execution; enables a user on the client system to execute a command on the remote system. The first command from client (`rcmd`) to server (`rshd`) uses well-known TCP port 512. A second TCP connection can be opened at the request of `rcmd`. The client port number for the second connection is sent to the server as an ASCII string.

- Login—Better known as `rlogin`; uses well-known TCP port 513. For details, see RFC 1282. No special firewall processing is required.

- Shell—Remote command execution; enables a user on the client system to execute a command on the remote system. The first command from client (`rcmd`) to server (`rshd`) uses well-known TCP port 514. A second TCP connection can be opened at the request of rcmd. The client port number for the second connection is sent to the server as an ASCII string.

Support of stateful firewall services requires that you configure the Exec ALG on TCP port 512, the Login ALG on TCP port 513, and the Shell ALG on TCP port 514. NAT remote-shell services require that any dynamic source port assigned be within the port range 512 to 1023. If you configure a NAT pool, this port range is reserved exclusively for remote shell applications.

WinFrame

WinFrame application server software provides access to virtually any Windows application, across any type of network connection to any type of client.

This protocol is mainly used by Citrix Windows applications.

Stateful firewall and NAT require the ALG Winframe to be configured on TCP destination port 1494 and UDP port 1604.

Juniper Networks Defaults

The Junos OS provides a default, hidden configuration group called junos-defaults that is automatically applied to the configuration of your router. The junos-defaults group contains preconfigured statements that contain predefined values for common applications. Some of the statements must be referenced to take effect, such as applications like FTP or Telnet. Other statements are applied automatically, such as terminal settings. All of the preconfigured statements begin with the reserved name junos-.

NOTE: You can override the Junos OS default configuration values, but you cannot delete or edit them. If you delete a configuration, the defaults return when a new configuration is added.

You cannot use the apply-groups statement with the Junos OS defaults group.

To view the full set of available preset statements from the Junos OS default group, issue the show groups junos-defaults configuration mode command. The following example displays the list of Junos OS default groups that use application protocols (ALGs):

```
user@host# show groups junos-defaults
applications {
    #
    # File Transfer Protocol
    #
    application junos-ftp {
        application-protocol ftp;
        protocol tcp;
        destination-port 21;
    }
    #
    # Trivial File Transfer Protocol
    #
    application junos-tftp {
        application-protocol tftp;
        protocol udp;
        destination-port 69;
    }
    #
    # RPC portmapper on TCP
    #
    application junos-rpc-portmap-tcp {
        application-protocol rpc-portmap;
        protocol tcp;
        destination-port 111;
    }
    #
    # RPC portmapper on UDP
    #
    application junos-rpc-portmap-udp {
        application-protocol rpc-portmap;
        protocol udp;
        destination-port 111;
    }
    #
    # SNMP get
    #
    application junos-snmp-get {
        application-protocol snmp;
        protocol udp;
        destination-port 161;
        snmp-command get;
    }
    #
    # SNMP get next
    #
    application junos-snmp-get-next {
        application-protocol snmp;
        protocol udp;
        destination-port 161;
```

```
        snmp-command get-next;
    }
    #
    # SNMP response
    #
    application junos-snmp-response {
        application-protocol snmp;
        protocol udp;
        source-port 161;
        snmp-command get-response;
    }
    #
    # SNMP trap
    #
    application junos-snmp-trap {
        application-protocol snmp;
        protocol udp;
        destination-port 162;
        snmp-command trap;
    }
    #
    # remote exec
    #
    application junos-rexec {
        application-protocol exec;
        protocol tcp;
        destination-port 512;
    }
    #
    # remote login
    #
    application junos-rlogin {
        application-protocol shell;
        protocol tcp;
        destination-port 513;
    }
    #
    # remote shell
    #
    application junos-rsh {
        application-protocol shell;
        protocol tcp;
        destination-port 514;
```

```
}
#
# Real Time Streaming Protocol
#
application junos-rtsp {
    application-protocol rtsp;
    protocol tcp;
    destination-port 554;
}
#
# Citrix windows application server protocol
# windows applications remotely on windows/non-windows clients
#
# citrix needs udp 1604 to be open
#
application junos-citrix-winframe {
    application-protocol winframe;
    protocol tcp;
    destination-port 1494;
}
application junos-citrix-winframe-udp {
    protocol udp;
    destination-port 1604;
}
#
# Oracle SQL servers use this protocol to execute sql commands
# from clients, load balance, use application-specific servers, etc
#
application junos-sqlnet {
    application-protocol sqlnet;
    protocol tcp;
    destination-port 1521;
}
#
# H.323 Protocol for audio/video conferencing
#
application junos-h323 {
    application-protocol h323;
    protocol tcp;
    destination-port 1720;
}
application junos-h323-ras {
    application-protocol ras;
```

```
        protocol udp;
        destination-port 1719;
    }
    #
    # Internet Inter-ORB Protocol - used for CORBA applications
    # The ORB protocol in Java virtual machines uses port 1975 as default
    #
    application junos-iiop-java {
        application-protocol iiop;
        protocol tcp;
        destination-port 1975;
    }
    #
    # Internet Inter-ORB Protocol - used for CORBA applications
    # ORBIX is a CORBA framework from Iona Technologies that uses port
    # 3075 as default
    #
    application junos-iiop-orbix {
        application-protocol iiop;
        protocol tcp;
        destination-port 3075;
    }
    #
    # Real players use this protocol for real time streaming
    # This was the original protocol for real players.
    # RTSP is more widely used by real players
    # but they still support realaudio.
    #
    application junos-realaudio {
        application-protocol realaudio;
        protocol tcp;
        destination-port 7070;
    }
    #
    # traceroute application.
    #
    application junos-traceroute {
        application-protocol traceroute;
        protocol udp;
        destination-port 33435-33450;
        ttl-threshold 30;
    }
    #
```

```
# The full range of known RPC programs using UDP
# The program numbers can be more specific to certain applications.
#
application junos-rpc-services-udp {
    application-protocol rpc;
    protocol udp;
    rpc-program-number 100000-400000;
}
#
# The full range of known RPC programs using TCP
# The program numbers can be more specific to certain applications.
#
application junos-rpc-services-tcp {
    application-protocol rpc;
    protocol tcp;
    rpc-program-number 100000-400000;
}
#
# All ICMP traffic
# This can be made to be more restrictive by specifying ICMP type
# and code.
#
application junos-icmp-all {
    application-protocol icmp;
}
#
#  Protocol used by Windows media server and windows media player
#
application junos-netshow {
    application-protocol netshow;
    protocol tcp;
    destination-port 1755;
}
#
# NetBIOS  - networking protocol used on
# Windows networks   name service port, both UDP and TCP
#
application junos-netbios-name-udp {
    application-protocol netbios;
    protocol udp;
    destination-port 137;
}
application junos-netbios-name-tcp {
        protocol tcp;
        destination-port 137;
    }
    #
    #  NetBIOS  - networking protocol used on
    # Windows networks   datagram service port
    #
    application junos-netbios-datagram {
        application-protocol netbios;
        protocol udp;
        destination-port 138;
    }
    #
    #  NetBIOS  - networking protocol used on
    # Windows networks   session service port
    #
    application junos-netbios-session {
        protocol tcp;
        destination-port 139;
    }
    #
    # DCE-RPC portmapper on TCP
    #
    application junos-dce-rpc-portmap {
        application-protocol dce-rpc-portmap;
        protocol tcp;
        destination-port 135;
    }
    #
    # DCE-RPC application on TCP  sample UUID
    # This application requires user to specify the UUID value
#
    # application junos-dcerpc {
        #   application-protocol dce-rpc;
        #   protocol tcp;
        #
        #   # UUID also needs to be defined as shown below
        #  UUID   11223344 22334455 33445566 44556677;
        #
        # }
        #
        # ms-exchange needs these 3 UUIDs
        #
        application junos-dcerpc-endpoint-mapper-service {
            application-protocol dce-rpc;
            protocol tcp;
            uuid e1af8308-5d1f-11c9-91a4-08002b14a0fa;
        }
        application junos-dcerpc-msexchange-directory-rfr {
            application-protocol dce-rpc;
            protocol tcp;
            uuid 1544f5e0-613c-11d1-93df-00c04fd7bd09;
        }
        application junos-dcerpc-msexchange-information-store {
            application-protocol dce-rpc;
            protocol tcp;
            uuid a4f1db00-ca47-1067-b31f-00dd010662da;
        }
        application junos-ssh {
            protocol tcp;
            destination-port 22;
        }
        application junos-telnet {
            protocol tcp;
            destination-port 23;
        }
        application junos-smtp {
            protocol tcp;
            destination-port 25;
        }
        application junos-dns-udp {
            protocol udp;
            destination-port 53;
        }
        application junos-dns-tcp {
            protocol tcp;
            destination-port 53;
        }
        application junos-tacacs {
            protocol tcp;
            destination-port 49;
        }
        # TACACS Database Service
        application junos-tacacs-ds {
            protocol tcp;
            destination-port 65;
        }
        application junos-dhcp-client {
            protocol udp;
            destination-port 68;
        }
        application junos-dhcp-server {
            protocol udp;
            destination-port 67;
        }
        application junos-bootpc {
            protocol udp;
            destination-port 68;
        }
        application junos-bootps {
            protocol udp;
            destination-port 67;
        }
        application junos-finger {
            protocol tcp;
            destination-port 79;
        }
        application junos-http {
            protocol tcp;
            destination-port 80;
        }
        application junos-https {
            protocol tcp;
            destination-port 443;
        }
        application junos-pop3 {
            protocol tcp;
            destination-port 110;
        }
        application junos-ident {
            protocol tcp;
            destination-port 113;
        }
        application junos-nntp {
            protocol tcp;
            destination-port 119;
        }
        application junos-ntp {
            protocol udp;
            destination-port 123;
        }
        application junos-imap {
            protocol tcp;
            destination-port 143;
        }
        application junos-imaps {
            protocol tcp;
            destination-port 993;
        }
        application junos-bgp {
            protocol tcp;
            destination-port 179;
        }
        application junos-ldap {
            protocol tcp;
            destination-port 389;
        }
        application junos-snpp {
            protocol tcp;
            destination-port 444;
        }
        application junos-biff {
            protocol udp;
            destination-port 512;
        }
        # UNIX who
        application junos-who {
            protocol udp;
            destination-port 513;
        }
        application junos-syslog {
            protocol udp;
            destination-port 514;
        }
        # line printer daemon, printer, spooler
        application junos-printer {
            protocol tcp;
            destination-port 515;
        }
        # UNIX talk
        application junos-talk-tcp {
            protocol tcp;
            destination-port 517;
        }
        application junos-talk-udp {
            protocol udp;
            destination-port 517;
        }
        application junos-ntalk {
            protocol udp;
            destination-port 518;
        }
        application junos-rip {
            protocol udp;
            destination-port 520;
        }
        # INA sanctioned RADIUS port numbers
        application junos-radius {
            protocol udp;
            destination-port 1812;
        }
        application junos-radacct {
            protocol udp;
            destination-port 1813;
        }
        application junos-nfsd-tcp {
            protocol tcp;
            destination-port 2049;
        }
        application junos-nfsd-udp {
            protocol udp;
            destination-port 2049;
        }
        application junos-cvspserver {
            protocol tcp;
            destination-port 2401;
        }
        #
        # Label Distribution Protocol
        #
        application junos-ldp-tcp {
            protocol tcp;
            destination-port 646;
        }
        application junos-ldp-udp {
            protocol udp;
            destination-port 646;
        }
        #
        # JUNOScript and JUNOScope management
        #
        application junos-xnm-ssl {
            protocol tcp;
            destination-port 3220;
        }
        application junos-xnm-clear-text {
            protocol tcp;
            destination-port 3221;
        }
        #
        # IPsec tunnel
        #
        application junos-ipsec-esp {
            protocol esp;
        }
        #
        #IKE application for IPSec VPN
        #
        application junos-ike {
            application-protocol ike-esp-nat;
            protocol udp;
            destination-port 500;
        }
        #
        # 'junos-algs-outbound' defines a set of all applications
        # requiring an ALG.  Useful for defining rule to the the public
        # internet allowing private network users to use all JUNOS OS
        # supported ALGs initiated from the private network.
        #
        # NOTE: the contents of this set might grow in future JUNOS OS versions.
        #
        application-set junos-algs-outbound {
            application junos-ftp;
            application junos-tftp;
            application junos-rpc-portmap-tcp;
            application junos-rpc-portmap-udp;
            application junos-snmp-get;
            application junos-snmp-get-next;
            application junos-snmp-response;
            application junos-snmp-trap;
            application junos-rexec;
            application junos-rlogin;
            application junos-rsh;
            application junos-rtsp;
            application junos-citrix-winframe;
            application junos-citrix-winframe-udp;
            application junos-sqlnet;
            application junos-h323;
            application junos-iiop-java;
            application junos-iiop-orbix;
            application junos-realaudio;
            application junos-traceroute;
            application junos-rpc-services-udp;
            application junos-rpc-services-tcp;
            application junos-icmp-all;
            application junos-netshow;
            application junos-netbios-name-udp;
            application junos-netbios-datagram;
            application junos-dcerpc-endpoint-mapper-service;
            application junos-dcerpc-msexchange-directory-rfr;
            application junos-dcerpc-msexchange-information-store;
        }
        #
        # 'junos-management-inbound' represents the group of applications
        # that might need access the router from public network for
        # for management purposes.
        #
        # Set is intended for a UI to display management choices.
        #
        # NOTE:  It is not recommended the user to use the entire set
        #        directly in a firewall rule and open up firewall to all
        #        of these applications.  Also, the user  should always
        #        specify the source and destination prefixes when using
        #        each application.
        #
        # NOTE: the contents of this set may grow in future JUNOS versions.
        #
        application-set junos-management-inbound {
            application junos-snmp-get;
            application junos-snmp-get-next;
            application junos-snmp-response;
```

```
            application junos-snmp-trap;
            application junos-ssh;
            application junos-telnet;
            application junos-http;
            application junos-https;
            application junos-xnm-ssl;
            application junos-xnm-clear-text;
        }
        #
        # 'junos-routing-inbound' represents routing protocols that might
        #  need to access the router from public network.
        #
        # Set is intended for a UI to display routing involvement choices.
        #
        # NOTE:  It is not recommended the user to use the entire set
        #        directly in a firewall rule and open up firewall to all
        #        of these applications.  Also, the user  should always
        #        specify the source and destination prefixes when using
        #        each application.
        #
        # NOTE: the contents of this set might grow in future JUNOS OS versions.
        #
        application-set junos-routing-inbound {
            application junos-bgp;
            application junos-rip;
            application junos-ldp-tcp;
            application junos-ldp-udp;
        }
        application-set junos-h323-suite {
            application junos-h323-ras,
            application junos-h323;
        }
    }
```

To reference statements available from the `junos-defaults` group, include the selected `junos-`*`default-name`* statement at the applicable hierarchy level. To configure application protocols, see "Configuring Application Properties" on page 508 ; for details about a specific protocol, see "ALG Descriptions" on page 475 .

Examples: Referencing the Preset Statement from the Junos OS Default Group

The following example is a preset statement from the Junos OS default groups that is available for FTP in a stateful firewall:

```
[edit]
groups {
    junos-defaults {
        applications {
            application junos-ftp { # Use FTP default configuration
            application-protocol ftp;
            protocol tcp;
            destination-port 21;
        }
    }
}
```

To reference a preset Junos OS default statement from the Junos OS default groups, include the `junos-`*`default-name`* statement at the applicable hierarchy level. For example, to reference the Junos OS default statement for FTP in a stateful firewall, include the `junos-ftp` statement at the `[edit services stateful-firewall rule` *`rule-name`* `term` *`term-name`* `from applications]` hierarchy level.

```
[edit]
services {
    stateful-firewall {
        rule my-rule {
            term my-term {
                from {
                    applications junos-ftp; #Reference predefined statement, junos-ftp,
                }
            }
        }
    }
}
```

The following example shows configuration of the default Junos IP ALG:

```
[edit]
services {
    stateful-firewall {
        rule r1 {
            match-direction input;
            term t1 {
                from {
                    applications junos-ip;
                }
                then {
                    accept;
                    syslog;
                }
            }
        }
    }
}
```

If you configure the IP ALG in the stateful firewall rule, it is matched by any IP traffic, but when any other more specific application matches the same traffic, the IP ALG is not matched. For example, in the following configuration, both the ICMP ALG and the IP ALG are configured, but traffic is matched for ICMP packets, because it is the more specific match.

```
[edit]
services {
    stateful-firewall {
        rule r1 {
            match-direction input;
            term t1 {
                from {
                    applications [ junos-ip junos-icmp-all ];
                }
                then {
                    accept;
                    syslog;
                }
            }
        }
    }
}
```

SEE ALSO

Configuring Application Sets | 532

ICMP, Ping, and Traceroute ALGs for MS-MICs and MS-MPCs

Starting with Junos OS Release 14.2, Junos OS extension-provider packages that are preinstalled and preconfigured on the MS-MIC and MS-MPC offer support for ping, traceroute, and ICMP ALGs in a consistent manner that is identical to the support that the uKernel service provides. Parity and uniformity of support is established for these ALGs between MS-MICs/MS-MPCs and the uKernel service. Until Junos OS Release 14.1, ICMP ALGs, ping ALGs, and traceroute ALGs were not entirely supported on MX Series routers with MS-MICs and MS-MPCs in comparison with the uKernel service that enables Network Address Translation (NAT) with stateful firewall (SFW) on the uKernel PIC. Support was available for handling of ICMP error response packets that match any existing flow in the opposite direction and NAT processing of ICMP packets with NAT processing of ping packets.

On MX Series routers with MS-MICs and MS-MPCs, tracking of ping traffic states wholly using the ICMP sequence numbers (for example, forwarding an echo reply only if the echo request with the corresponding sequence number is identified) is supported. ICMP application layer gateway (ALG) is enhanced to provide detailed logging information. Also, the traceroute ALGs enable UDP probe packets to be processed with the UDP destination port number greater than 33000 and the IP time-to-live (TTL) is less than 30 seconds. Traceroute ALGs enable ICMP response packets for which the ICMP type is time-exceeded to be processed and support a traceroute TTL threshold value, which controls the acceptable level of network penetration for trace routing.

You can configure ICMP and ping messages with the `application junos-icmp-all`, `application junos-icmp-ping`, and `application icmp-code` statements at the `[edit services stateful-firewall rule` *`rule-name`* `term` *`term-name`* `from]` and the `[edit services nat rule` *`rule-name`* `term` *`term-name`* `from]` hierarchy levels to define the match condition for the stateful firewall and NAT rules. Until Junos OS Release 14.1, a restriction or a validation on the applications that you could define for ICMP messages was not present. MS-MICs and MS-MPCs function the same way as the uKernel service, which causes the ping traffic to be tracked statefully using the ICMP sequence numbers (an echo reply is forwarded only if echo request with the corresponding sequence number matches). Also, MS-MICs and MS-MPCs impose a limit on the outstanding ping requests and drop the subsequent ping requests when the limit is reached.

Similarly, for traceroute messages, you can configure the `application junos-traceroute` and `application junos-traceroute-ttl-1` statements at the `[edit services stateful-firewall rule` *`rule-name`* `term` *`term-name`* `from]` and the `[edit services nat rule` *`rule-name`* `term` *`term-name`* `from]` hierarchy levels to define the match condition for traceroute messages for the stateful firewall and NAT rules.

Traceroute and ICMP messages are supported for both IPv4 and IPv6 packets. For the traceroute functionality to work, you only need to ensure that the user-defined applications are working as expected with the inactivity timeout period and the TTL threshold values are configured to be the same period of time as configured by using the `session-timeout` *`seconds`* statement at the `[edit services application-identification application` *`application-name`*`]` hierarchy level. During the logging of ICMP messages, the ALG information for ping and ICMP utilities is displayed in the output of the relevant show commands, such as `show sessions` and `show conversations`, in the same manner as displayed for uKernel logging.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
18.2R1	SIP supported for DS-Lite on MS-MPC and MS-MIC starting in Junos OS Release 18.2R1.
18.1R1	Support for ALGs with DS-lite on MS-MPC and MS-MIC starts in Junos OS Release 18.1R1
17.2R1	(Starting in Junos OS Release 17.2R1)
17.2R1	(Starting in Junos OS Release 17.2R1)
17.2R1	Starting in Junos OS Release 17.2R1, NAT-64 rules are also supported.
17.2R1	Starting in Junos OS Release 17.2R1, NAT-64 rules are also supported.
17.1R1	(Starting in Junos OS Release 17.1R1)
17.1R1	Starting in Junos OS Release 17.1R1, the gatekeeper registration, administration, and status (RAS) ALG allows full support of gatekeeper mode for H.323 calls.
14.2R7	IKE ALG (Starting in Junos OS Release 14.2R7, 15.1R5, 16.1R2, and 17.1R1)
14.2R7	Starting in Junos OS Release 14.2R7, 15.1R5, 16.1R2, and 17.1R1, the IKE ALG enables the passing of IKEv1 and IPsec packets through NAPT-44 and NAT64 rules between IPsec peers that are not NAT-T compliant.

ALG Applications

IN THIS SECTION

- Configuring Application Properties | 508
- Configuring Application Sets | 532
- Examples: Configuring Application Protocols | 532
- Verifying the Output of ALG Sessions | 533

Configuring Application Properties

IN THIS SECTION

- Configuring an Application Protocol | 509
- Configuring the Network Protocol | 512
- Configuring the ICMP Code and Type | 514
- Configuring Source and Destination Ports | 515
- Configuring the Inactivity Timeout Period | 520
- Configuring an IKE ALG Application | 520
- Configuring SIP | 522
- Configuring an SNMP Command for Packet Matching | 531
- Configuring an RPC Program Number | 531
- Configuring the TTL Threshold | 531
- Configuring a Universal Unique Identifier | 532

To configure application properties, include the `application` statement at the `[edit applications]` hierarchy level:

```
[edit applications]
application application-name {
    application-protocol protocol-name;
    child-inactivity-timeout seconds;
    destination-port port-number;
    gate-timeout seconds;
    icmp-code value;
    icmp-type value;
    inactivity-timeout value;
    protocol type;
    rpc-program-number number;
    snmp-command command;
    source-port port-number;
    ttl-threshold value;
    uuid hex-value;
}
```

You can group application objects by configuring the application-set statement; for more information, see "Configuring Application Sets" on page 532 .

This section includes the following tasks for configuring applications:

Configuring an Application Protocol

The application-protocol statement allows you to specify which of the supported application protocols (ALGs) to configure and include in an application set for service processing. To configure application protocols, include the application-protocol statement at the [edit applications application *application-name*] hierarchy level:

```
[edit applications application application-name]
application-protocol protocol-name;
```

Table 18 on page 509 shows the list of supported protocols. For more information about specific protocols, see "ALG Descriptions" on page 475 .

Table 18: Application Protocols Supported by Services Interfaces

Protocol Name	CLI Value	Comments
Bootstrap protocol (BOOTP)	bootp	Supports BOOTP and dynamic host configuration protocol (DHCP).
Distributed Computing Environment (DCE) remote procedure call (RPC)	dce-rpc	Requires the protocol statement to have the value udp or tcp. Requires a uuid value. You cannot specify destination-port or source-port values.
DCE RPC portmap	dce-rpc-portmap	Requires the protocol statement to have the value udp or tcp. Requires a destination-port value.
Domain Name System (DNS)	dns	Requires the protocol statement to have the value udp. This application protocol closes the DNS flow as soon as the DNS response is received.
Exec	exec	Requires the protocol statement to have the value tcp or to be unspecified. Requires a destination-port value.

Table 18: Application Protocols Supported by Services Interfaces *(Continued)*

Protocol Name	CLI Value	Comments
FTP	`ftp`	Requires the `protocol` statement to have the value `tcp` or to be unspecified. Requires a `destination-port` value.
H.323	`h323`	–
IKE ALG	`ike-esp-nat`	Requires the `protocol` statement to have the value `udp` and requires the `destination-port` value to be 500.
Internet Control Message Protocol (ICMP)	`icmp`	Requires the `protocol` statement to have the value `icmp` or to be unspecified.
Internet Inter-ORB Protocol	`iiop`	–
IP	`ip`	–
Login	`login`	–
NetBIOS	`netbios`	Requires the `protocol` statement to have the value `udp` or to be unspecified. Requires a `destination-port` value.
NetShow	`netshow`	Requires the `protocol` statement to have the value `tcp` or to be unspecified. Requires a `destination-port` value.
Point-to-Point Tunneling Protocol	`pptp`	–
RealAudio	`realaudio`	–
Real-Time Streaming Protocol (RTSP)	`rtsp`	Requires the `protocol` statement to have the value `tcp` or to be unspecified. Requires a `destination-port` value.

Table 18: Application Protocols Supported by Services Interfaces *(Continued)*

Protocol Name	CLI Value	Comments
RPC User Datagram Protocol (UDP) or TCP	`rpc`	Requires the `protocol` statement to have the value `udp` or `tcp`. Requires a `rpc-program-number` value. You cannot specify `destination-port` or `source-port` values.
RPC port mapping	`rpc-portmap`	Requires the `protocol` statement to have the value `udp` or `tcp`. Requires a `destination-port` value.
Shell	`shell`	Requires the `protocol` statement to have the value `tcp` or to be unspecified. Requires a `destination-port` value.
Session Initiation Protocol	`sip`	–
SNMP	`snmp`	Requires the `protocol` statement to have the value `udp` or to be unspecified. Requires a `destination-port` value.
SQLNet	`sqlnet`	Requires the `protocol` statement to have the value `tcp` or to be unspecified. Requires a `destination-port` or `source-port` value.
Talk Program	`talk`	
Trace route	`traceroute`	Requires the `protocol` statement to have the value `udp` or to be unspecified. Requires a `destination-port` value.
Trivial FTP (TFTP)	`tftp`	Requires the `protocol` statement to have the value `udp` or to be unspecified. Requires a `destination-port` value.
WinFrame	`winframe`	–

NOTE: You can configure application-level gateways (ALGs) for ICMP and trace route under stateful firewall, NAT, or CoS rules when twice NAT is configured in the same service set. These ALGs cannot be applied to flows created by the Packet Gateway Controller Protocol (PGCP).

Twice NAT does not support any other ALGs. NAT applies only the IP address and TCP or UDP headers, but not the payload.

For more information about configuring twice NAT, see "Junos Address Aware Network Addressing Overview" on page 53 .

Configuring the Network Protocol

The protocol statement allows you to specify which of the supported network protocols to match in an application definition. To configure network protocols, include the protocol statement at the [edit applications application *application-name*] hierarchy level:

```
[edit applications application application-name]
protocol type;
```

You specify the protocol type as a numeric value; for the more commonly used protocols, text names are also supported in the command-line interface (CLI). Table 19 on page 512 shows the list of the supported protocols.

Table 19: Network Protocols Supported by Services Interfaces

Network Protocol Type	CLI Value	Comments
IP Security (IPsec) authentication header (AH)	ah	–
External Gateway Protocol (EGP)	egp	–
IPsec Encapsulating Security Payload (ESP)	esp	–
Generic routing encapsulation (GR)	gre	–
ICMP	icmp	Requires an application-protocol value of icmp.
ICMPv6	icmp6	Requires an application-protocol value of icmp.

Table 19: Network Protocols Supported by Services Interfaces *(Continued)*

Network Protocol Type	CLI Value	Comments
Internet Group Management Protocol (IGMP)	`igmp`	–
IP in IP	`ipip`	–
OSPF	`ospf`	–
Protocol Independent Multicast (PIM)	`pim`	–
Resource Reservation Protocol (RSVP)	`rsvp`	–
TCP	`tcp`	Requires a `destination-port` or `source-port` value unless you specify `application-protocol rcp` or `dce-rcp`.
UDP	`udp`	Requires a `destination-port` or `source-port` value unless you specify `application-protocol rcp` or `dce-rcp`.

For a complete list of possible numeric values, see RFC 1700, *Assigned Numbers (for the Internet Protocol Suite)*.

NOTE: IP version 6 (IPv6) is not supported as a network protocol in application definitions.

By default, the twice NAT feature can affect IP, TCP, and UDP headers embedded in the payload of ICMP error messages. You can include the `protocol tcp` and `protocol udp` statements with the application statement for twice NAT configurations. For more information about configuring twice NAT, see "Junos Address Aware Network Addressing Overview" on page 53 .

Configuring the ICMP Code and Type

The ICMP code and type provide additional specification, in conjunction with the network protocol, for packet matching in an application definition. To configure ICMP settings, include the icmp-code and icmp-type statements at the [edit applications application *application-name*] hierarchy level:

```
[edit applications application application-name]
icmp-code value;
icmp-type value;
```

You can include only one ICMP code and type value. The application-protocol statement must have the value icmp. Table 20 on page 514 shows the list of supported ICMP values.

Table 20: ICMP Codes and Types Supported by Services Interfaces

CLI Statement	Description
icmp-code	This value or keyword provides more specific information than icmp-type. Because the value's meaning depends upon the associated icmp-type value, you must specify icmp-type along with icmp-code. For more information, see the Routing Policies, Firewall Filters, and Traffic Policers User Guide. In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed). The keywords are grouped by the ICMP type with which they are associated: parameter-problem: ip-header-bad (0), required-option-missing (1) redirect: redirect-for-host (1), redirect-for-network (0), redirect-for-tos-and-host (3), redirect-for-tos-and-net (2) time-exceeded: ttl-eq-zero-during-reassembly (1), ttl-eq-zero-during-transit (0) unreachable: communication-prohibited-by-filtering (13), destination-host-prohibited (10), destination-host-unknown (7), destination-network-prohibited (9), destination-network-unknown (6), fragmentation-needed (4), host-precedence-violation (14), host-unreachable (1), host-unreachable-for-TOS (12), network-unreachable (0), network-unreachable-for-TOS (11), port-unreachable (3), precedence-cutoff-in-effect (15), protocol-unreachable (2), source-host-isolated (8), source-route-failed (5)

Table 20: ICMP Codes and Types Supported by Services Interfaces *(Continued)*

CLI Statement	Description
icmp-type	Normally, you specify this match in conjunction with the protocol match statement to determine which protocol is being used on the port. For more information, see the Routing Policies, Firewall Filters, and Traffic Policers User Guide. In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed): echo-reply (0), echo-request (8), info-reply (16), info-request (15), mask-request (17), mask-reply (18), parameter-problem (12), redirect (5), router-advertisement (9), router-solicit (10), source-quench (4), time-exceeded (11), timestamp (13), timestamp-reply (14), or unreachable (3).

NOTE: If you configure an interface with an input firewall filter that includes a reject action and with a service set that includes stateful firewall rules, the router executes the input firewall filter before the stateful firewall rules are run on the packet. As a result, when the Packet Forwarding Engine sends an ICMP error message out through the interface, the stateful firewall rules might drop the packet because it was not seen in the input direction.

Possible workarounds are to include a forwarding-table filter to perform the reject action, because this type of filter is executed after the stateful firewall in the input direction, or to include an output service filter to prevent the locally generated ICMP packets from going to the stateful firewall service.

Configuring Source and Destination Ports

The TCP or UDP source and destination port provide additional specification, in conjunction with the network protocol, for packet matching in an application definition. To configure ports, include the destination-port and source-port statements at the [edit applications application *application-name*] hierarchy level:

```
[edit applications application application-name]
destination-port value;
source-port value;
```

You must define one source or destination port. Normally, you specify this match in conjunction with the protocol match statement to determine which protocol is being used on the port; for constraints, see Table 18 on page 509 .

You can specify either a numeric value or one of the text synonyms listed in Table 21 on page 516 .

Table 21: Port Names Supported by Services Interfaces

Port Name	Corresponding Port Number
afs	1483
bgp	179
biff	512
bootpc	68
bootps	67
cmd	514
cvspserver	2401
dhcp	67
domain	53
eklogin	2105
ekshell	2106
exec	512
finger	79
ftp	21

Table 21: Port Names Supported by Services Interfaces *(Continued)*

Port Name	Corresponding Port Number
ftp-data	20
http	80
https	443
ident	113
imap	143
kerberos-sec	88
klogin	543
kpasswd	761
krb-prop	754
krbupdate	760
kshell	544
ldap	389
login	513
mobileip-agent	434
mobilip-mn	435

Table 21: Port Names Supported by Services Interfaces ***(Continued)***

Port Name	Corresponding Port Number
msdp	639
netbios-dgm	138
netbios-ns	137
netbios-ssn	139
nfsd	2049
nntp	119
ntalk	518
ntp	123
pop3	110
pptp	1723
printer	515
radacct	1813
radius	1812
rip	520
rkinit	2108

Table 21: Port Names Supported by Services Interfaces *(Continued)*

Port Name	Corresponding Port Number
smtp	25
snmp	161
snmptrap	162
snpp	444
socks	1080
ssh	22
sunrpc	111
syslog	514
tacacs-ds	65
talk	517
telnet	23
tftp	69
timed	525
who	513
xdmcp	177

Table 21: Port Names Supported by Services Interfaces *(Continued)*

Port Name	Corresponding Port Number
zephyr-clt	2103
zephyr-hm	2104

For more information about matching criteria, see the Routing Policies, Firewall Filters, and Traffic Policers User Guide.

Configuring the Inactivity Timeout Period

You can specify a timeout period for application inactivity. If the software has not detected any activity during the duration, the flow becomes invalid when the timer expires. To configure a timeout period, include the `inactivity-timeout` statement at the `[edit applications application application-name]` hierarchy level:

```
[edit applications application application-name]
inactivity-timeout seconds;
```

The default value is 30 seconds. The value you configure for an application overrides any global value configured at the `[edit interfaces interface-name service-options]` hierarchy level; for more information, see *Configuring Default Timeout Settings for Services Interfaces*.

Configuring an IKE ALG Application

Before Junos OS Release 17.4R1, Network Address Translation-Traversal (NAT-T) is not supported for the Junos VPN Site Secure suite of IPsec features on the MX Series routers. The IKE ALG enables the passing of IKEv1 and IPsec packets through NAPT-44 and NAT64 filters between IPsec peers that are not NAT-T compliant. This ALG supports only ESP tunnel mode. You can use the predefined IKE ALG application `junos-ike`, which has predefined values for the destination port (500), inactivity timeout (30 seconds), gate timeout (120 seconds), and ESP session idle timeout (800 seconds). If you want to use the IKE ALG with values different from the predefined `junos-ike` application, you need to configure a new IKE ALG application.

To configure an IKE ALG application:

1. Specify a name for the application.

```
[edit applications]
user@host# set application junos-ike
```

2. Specify the IKE ALG.

```
[edit applications application junos-ike]
user@host# set application-protocol ike-esp-nat
```

3. Specify the UDP protocol.

```
[edit applications application junos-ike]
user@host# set protocol udp
```

4. Specify 500 for the destination port.

```
[edit applications application junos-ike]
user@host# set destination-port 500
```

5. Specify the number of seconds that the IKE session is inactive before it is deleted. The default is 30 seconds.

```
[edit applications application junos-ike]
user@host# set inactivity-timeout seconds
```

6. Specify the number of seconds that can pass after IKE establishes the security association between the IPsec client and server and before the ESP traffic starts in both directions. If the ESP traffic has not started before this timeout value, the ESP gates are deleted and the ESP traffic is blocked. The default is 120 seconds.

```
[edit applications application junos-ike]
user@host# set gate-timeout seconds
```

7. Specify the ESP session (IPsec data traffic) idle timeout in seconds. If no IPsec data traffic is passed on the ESP session in this time, the session is deleted. The default is 800 seconds.

   ```
   [edit applications application junos-ike]
   user@host# set child-inactivity-timeout seconds
   ```

Configuring SIP

The Session Initiation Protocol (SIP) is a generalized protocol for communication between endpoints involved in Internet services such as telephony, fax, video conferencing, instant messaging, and file exchange.

The Junos OS provides ALG services in accordance with the standard described in RFC 3261, *SIP: Session Initiation Protocol*. SIP flows under the Junos OS are as described in RFC 3665, *Session Initiation Protocol (SIP) Basic Call Flow Examples*.

> **NOTE**: Before implementing the Junos OS SIP ALG, you should be familiar with certain limitations, discussed in "Junos OS SIP ALG Limitations" on page 530
>
> The use of NAT in conjunction with the SIP ALG results in changes in SIP header fields due to address translation. For an explanation of these translations, refer to "SIP ALG Interaction with Network Address Translation" on page 523 .

To implement SIP on adaptive services interfaces, you configure the `application-protocol` statement at the `[edit applications application application-name]` hierarchy level with the value `sip`. For more information about this statement, see "Configuring an Application Protocol" on page 509 . In addition, there are two other statements you can configure to modify how SIP is implemented:

- You can enable the router to accept any incoming SIP calls for the endpoint devices that are behind the NAT firewall. When a device behind the firewall registers with the proxy that is outside the firewall, the AS or Multiservices PIC maintains the registration state. When the `learn-sip-register` statement is enabled, the router can use this information to accept inbound calls. If this statement is not configured, no inbound calls are accepted; only the devices behind the firewall can call devices outside the firewall.

  To configure SIP registration, include the `learn-sip-register` statement at the `[edit applications application application-name]` hierarchy level:

  ```
  [edit applications application application-name]
  learn-sip-register;
  ```

NOTE: The learn-sip-register statement is not applicable to the Next Gen Services MX-SPC3.

You can also manually inspect the SIP register by issuing the show services stateful-firewall sip-register command; for more information, see the *Junos OS System Basics and Services Command Reference*. The show services stateful-firewall sip-register command is not supported for Next Gen Services.

- You can specify a timeout period for the duration of SIP calls that are placed on hold. When a call is put on hold, there is no activity and flows might time out after the configured inactivity-timeout period expires, resulting in call state teardown. To avoid this, when a call is put on hold, the flow timer is reset to the sip-call-hold-timeout cycle to preserve the call state and flows for longer than the inactivity-timeout period.

  NOTE: The sip-call-hold-timeout statement is not applicable to the Next Gen Services MX-SPC3.

  To configure a timeout period, include the sip-call-hold-timeout statement at the [edit applications application *application-name*] hierarchy level:

  ```
  [edit applications application application-name]
  sip-call-hold-timeout seconds;
  ```

  The default value is 7200 seconds and the range is from 0 through 36,000 seconds (10 hours).

SIP ALG Interaction with Network Address Translation

The Network Address Translation (NAT) protocol enables multiple hosts in a private subnet to share a single public IP address to access the Internet. For outgoing traffic, NAT replaces the private IP address of the host in the private subnet with the public IP address. For incoming traffic, the public IP address is converted back into the private address, and the message is routed to the appropriate host in the private subnet.

Using NAT with the Session Initiation Protocol (SIP) service is more complicated because SIP messages contain IP addresses in the SIP headers as well as in the SIP body. When using NAT with the SIP service, the SIP headers contain information about the caller and the receiver, and the device translates this information to hide it from the outside network. The SIP body contains the Session Description Protocol (SDP) information, which includes IP addresses and port numbers for transmission of the media. The device translates SDP information for allocating resources to send and receive the media.

How IP addresses and port numbers in SIP messages are replaced depends on the direction of the message. For an outgoing message, the private IP address and port number of the client are replaced with the public IP address and port number of the Juniper Networks firewall. For an incoming message, the public address of the firewall is replaced with the private address of the client.

When an INVITE message is sent out across the firewall, the SIP Application Layer Gateway (ALG) collects information from the message header into a call table, which it uses to forward subsequent messages to the correct endpoint. When a new message arrives, for example an ACK or 200 OK, the ALG compares the “From:, To:, and Call-ID:” fields against the call table to identify the call context of the message. If a new INVITE message arrives that matches the existing call, the ALG processes it as a REINVITE.

When a message containing SDP information arrives, the ALG allocates ports and creates a NAT mapping between them and the ports in the SDP. Because the SDP requires sequential ports for the Real-Time Transport Protocol (RTP) and Real-Time Control Protocol (RTCP) channels, the ALG provides consecutive even-odd ports. If it is unable to find a pair of ports, it discards the SIP message.

This topic contains the following sections:

Outgoing Calls

When a SIP call is initiated with a SIP request message from the internal to the external network, NAT replaces the IP addresses and port numbers in the SDP and binds the IP addresses and port numbers to the Juniper Networks firewall. Via, Contact, Route, and Record-Route SIP header fields, if present, are also bound to the firewall IP address. The ALG stores these mappings for use in retransmissions and for SIP response messages.

The SIP ALG then opens pinholes in the firewall to allow media through the device on the dynamically assigned ports negotiated based on information in the SDP and the Via, Contact, and Record-Route header fields. The pinholes also allow incoming packets to reach the Contact, Via, and Record-Route IP addresses and ports. When processing return traffic, the ALG inserts the original Contact, Via, Route, and Record-Route SIP fields back into packets.

Incoming Calls

Incoming calls are initiated from the public network to public static NAT addresses or to interface IP addresses on the device. Static NATs are statically configured IP addresses that point to internal hosts; interface IP addresses are dynamically recorded by the ALG as it monitors REGISTER messages sent by internal hosts to the SIP registrar. When the device receives an incoming SIP packet, it sets up a session and forwards the payload of the packet to the SIP ALG.

The ALG examines the SIP request message (initially an INVITE) and, based on information in the SDP, opens gates for outgoing media. When a 200 OK response message arrives, the SIP ALG performs NAT on the IP addresses and ports and opens pinholes in the outbound direction. (The opened gates have a short time-to-live, and they time out if a 200 OK response message is not received quickly.)

When a 200 OK response arrives, the SIP proxy examines the SDP information and reads the IP addresses and port numbers for each media session. The SIP ALG on the device performs NAT on the addresses and port numbers, opens pinholes for outbound traffic, and refreshes the timeout for gates in the inbound direction.

When the ACK arrives for the 200 OK, it also passes through the SIP ALG. If the message contains SDP information, the SIP ALG ensures that the IP addresses and port numbers are not changed from the previous INVITE—if they are, the ALG deletes old pinholes and creates new pinholes to allow media to pass through. The ALG also monitors the Via, Contact, and Record-Route SIP fields and opens new pinholes if it determines that these fields have changed.

Forwarded Calls

A forwarded call is when, for example, user A outside the network calls user B inside the network, and user B forwards the call to user C outside the network. The SIP ALG processes the INVITE from user A as a normal incoming call. But when the ALG examines the forwarded call from B to C outside the network and notices that B and C are reached using the same interface, it does not open pinholes in the firewall, because media will flow directly between user A and user C.

Call Termination

The BYE message terminates a call. When the device receives a BYE message, it translates the header fields just as it does for any other message. But because a BYE message must be acknowledged by the receiver with a 200 OK, the ALG delays call teardown for five seconds to allow time for transmission of the 200 OK.

Call Re-INVITE Messages

Re-INVITE messages add new media sessions to a call and remove existing media sessions. When new media sessions are added to a call, new pinholes are opened in the firewall and new address bindings are created. The process is identical to the original call setup. When one or more media sessions are removed from a call, pinholes are closed and bindings released just as with a BYE message.

Call Session Timers

The SIP ALG uses the Session-Expires value to time out a session if a Re-INVITE or UPDATE message is not received. The ALG gets the Session-Expires value, if present, from the 200 OK response to the INVITE and uses this value for signaling timeout. If the ALG receives another INVITE before the session times out, it resets all timeout values to this new INVITE or to default values, and the process is repeated.

As a precautionary measure, the SIP ALG uses hard timeout values to set the maximum amount of time a call can exist. This ensures that the device is protected should one of the following events occur:

- End systems crash during a call and a BYE message is not received.
- Malicious users never send a BYE in an attempt to attack a SIP ALG.
- Poor implementations of SIP proxy fail to process Record-Route and never send a BYE message.
- Network failures prevent a BYE message from being received.

Call Cancellation

Either party can cancel a call by sending a CANCEL message. Upon receiving a CANCEL message, the SIP ALG closes pinholes through the firewall—if any have been opened—and releases address bindings. Before releasing the resources, the ALG delays the control channel age-out for approximately five seconds to allow time for the final 200 OK to pass through. The call is terminated when the five second timeout expires, regardless of whether a 487 or non-200 response arrives.

Forking

Forking enables a SIP proxy to send a single INVITE message to multiple destinations simultaneously. When the multiple 200 OK response messages arrive for the single call, the SIP ALG parses but updates call information with the first 200 OK messages it receives.

SIP Messages

The SIP message format consists of a SIP header section and the SIP body. In request messages, the first line of the header section is the request line, which includes the method type, request-URI, and protocol version. In response messages, the first line is the status line, which contains a status code. SIP headers contain IP addresses and port numbers used for signaling. The SIP body, separated from the header section by a blank line, is reserved for session description information, which is optional. Junos OS currently supports the SDP only. The SIP body contains IP addresses and port numbers used to transport the media.

SIP Headers

In the following sample SIP request message, NAT replaces the IP addresses in the header fields to hide them from the outside network.

```
INVITE bob@10.150.20.5 SIP/2.0
Via: SIP/2.0/UDP 10.150.20.3:5434
From: alice@10.150.20.3
To: bob@10.150.20.5
Call-ID: a12abcde@10.150.20.3
```

```
Contact: alice@10.150.20.3:5434
Route: <sip:netscreen@10.150.20.3:5060>
Record-Route: <sip:netscreen@10.150.20.3:5060>
```

How IP address translation is performed depends on the type and direction of the message. A message can be any of the following:

- Inbound request
- Outbound response
- Outbound request
- Inbound response

Table 22 on page 527 shows how NAT is performed in each of these cases. Note that for several of the header fields the ALG determine more than just whether the messages comes from inside or outside the network. It must also determine what client initiated the call, and whether the message is a request or response.

Table 22: Requesting Messages with NAT Table

Inbound Request (from public to private)	To:	Replace domain with local address
	From:	None
	Call-ID:	None
	Via:	None
	Request-URI:	Replace ALG address with local address
	Contact:	None
	Record-Route:	None
	Route:	None
Outbound Response	To:	Replace ALG address with local address

(from private to public)	From:	None
	Call-ID:	None
	Via:	None
	Request-URI:	N/A
	Contact:	Replace local address with ALG address
	Record-Route:	Replace local address with ALG address
	Route:	None
Outbound Request (from private to public)	To:	None
	From:	Replace local address with ALG address
	Call-ID:	None
	Via:	Replace local address with ALG address
	Request-URI:	None
	Contact:	Replace local address with ALG address
	Record-Route:	Replace local address with ALG address
	Route:	Replace ALG address with local address
Outbound Response (from public to private)	To:	None
	From:	Replace ALG address with local address

	Call-ID:	None
	Via:	Replace ALG address with local address
	Request-URI:	N/A
	Contact:	None
	Record-Route:	Replace ALG address with local address
	Route:	Replace ALG address with local address

SIP Body

The SDP information in the SIP body includes IP addresses the ALG uses to create channels for the media stream. Translation of the SDP section also allocates resources, that is, port numbers to send and receive the media.

The following excerpt from a sample SDP section shows the fields that are translated for resource allocation.

```
o=user 2344234 55234434 IN IP4 10.150.20.3
c=IN IP4 10.150.20.3
m=audio 43249 RTP/AVP 0
```

SIP messages can contain more than one media stream. The concept is similar to attaching multiple files to an e-mail message. For example, an INVITE message sent from a SIP client to a SIP server might have the following fields:

```
c=IN IP4 10.123.33.4
m=audio 33445 RTP/AVP 0
c=IN IP4 10.123.33.4
m=audio 33447 RTP/AVP 0
c=IN IP4 10.123.33.4
m=audio 33449 RTP/AVP 0
```

Junos OS supports up to 6 SDP channels negotiated for each direction, for a total of 12 channels per call.

Junos OS SIP ALG Limitations

The following limitations apply to configuration of the SIP ALG:

- Only the methods described in RFC 3261 are supported.
- Only SIP version 2 is supported.
- TCP is not supported as a transport mechanism for signaling messages for MS-MPCs but is supported for Next Gen Services.
- *Do not configure the SIP ALG when using STUN.* if clients use STUN/TURN to detect the firewall or NAT devices between the caller and responder or proxy, the client attempts to best-guess the NAT device behavior and act accordingly to place the call.
- On MS-MPCs, do not use the endpoint-independent mapping NAT pool option in conjunction with the SIP ALG. Errors will result. This does not apply to Next Gen Services.
- IPv6 signaling data is not supported for MS-MPCs but is supported for Next Gen Services.
- Authentication is not supported.
- Encrypted messages are not supported.
- SIP fragmentation is not supported for MS-MPCs but is supported for Next Gen Services.
- The maximum UDP packet size containing a SIP message is assumed to be 9 KB. SIP messages larger than this are not supported.
- The maximum number of media channels in a SIP message is assumed to be six.
- Fully qualified domain names (FQDNs) are not supported in critical fields.
- QoS is not supported. SIP supports DSCP rewrites.
- High availability is not supported, except for warm standby.
- A timeout setting of never is not supported on SIP or NAT.
- Multicast (forking proxy) is not supported.

Configuring an SNMP Command for Packet Matching

You can specify an SNMP command setting for packet matching. To configure SNMP, include the `snmp-command` statement at the `[edit applications application application-name]` hierarchy level:

```
[edit applications application application-name]
snmp-command value;
```

The supported values are `get`, `get-next`, `set`, and `trap`. You can configure only one value for matching. The `application-protocol` statement at the `[edit applications application application-name]` hierarchy level must have the value `snmp`. For information about specifying the application protocol, see "Configuring an Application Protocol" on page 509 .

Configuring an RPC Program Number

You can specify an RPC program number for packet matching. To configure an RPC program number, include the `rpc-program-number` statement at the `[edit applications application application-name]` hierarchy level:

```
[edit applications application application-name]
rpc-program-number number;
```

The range of values used for DCE or RPC is from 100,000 through 400,000. The `application-protocol` statement at the `[edit applications application application-name]` hierarchy level must have the value `rpc`. For information about specifying the application protocol, see "Configuring an Application Protocol" on page 509 .

Configuring the TTL Threshold

You can specify a trace route time-to-live (TTL) threshold value, which controls the acceptable level of network penetration for trace routing. To configure a TTL value, include the `ttl-threshold` statement at the `[edit applications application application-name]` hierarchy level:

```
[edit applications application application-name]
ttl-threshold value;
```

The `application-protocol` statement at the `[edit applications application application-name]` hierarchy level must have the value `traceroute`. For information about specifying the application protocol, see "Configuring an Application Protocol" on page 509 .

Configuring a Universal Unique Identifier

You can specify a Universal Unique Identifier (UUID) for DCE RPC objects. To configure a UUID value, include the uuid statement at the [edit applications application *application-name*] hierarchy level:

```
[edit applications application application-name]
uuid hex-value;
```

The uuid value is in hexadecimal notation. The application-protocol statement at the [edit applications application *application-name* hierarchy level must have the value dce-rpc. For information about specifying the application protocol, see "Configuring an Application Protocol" on page 509 . For more information on UUID numbers, see http://www.opengroup.org/onlinepubs/9629399/apdxa.htm.

SEE ALSO

ALGs Available for Junos OS Address Aware NAT | 71

Configuring Application Sets

You can group the applications you have defined into a named object by including the application-set statement at the [edit applications] hierarchy level with an application statement for each application:

```
[edit applications]
    application-set application-set-name {
    application application;
}
```

For an example of a typical application set, see "Examples: Configuring Application Protocols" on page 532 .

Examples: Configuring Application Protocols

The following example shows an application protocol definition describing a special FTP application running on port 78:

```
[edit applications]
application my-ftp-app {
    application-protocol ftp;
    protocol tcp;
    destination-port 78;
    timeout 100; # inactivity timeout for FTP service
}
```

The following example shows a special ICMP protocol (`application-protocol icmp`) of type 8 (ICMP echo):

```
[edit applications]
application icmp-app {
    application-protocol icmp;
    protocol icmp;
    icmp-type icmp-echo;
}
```

The following example shows a possible application set:

```
[edit applications]
application-set basic {
    http;
    ftp;
    telnet;
    nfs;
    icmp;
}
```

The software includes a predefined set of well-known application protocols. The set includes applications for which the TCP and UDP destination ports are already recognized by stateless firewall filters.

Verifying the Output of ALG Sessions

IN THIS SECTION

- FTP Example | 534
- RTSP ALG Example | 540
- System Log Messages | 543

This section contains examples of successful output from ALG sessions and information on system log configuration. You can compare the results of your sessions to check whether the configurations are functioning correctly.

FTP Example

This example analyzes the output during an active FTP session. It consists of four different flows; two are control flows and two are data flows. The example consists of the following parts:

Sample Output

MS-MPC Card

For MS-MPCs, the following is a complete sample output from the `show services stateful-firewall conversations application-protocol ftp` operational mode command:

```
user@host>show services stateful-firewall conversations application-protocol ftp
Interface: ms-1/3/0, Service set: CLBJI1-AAF001
Conversation: ALG protocol: ftp
  Number of initiators: 2, Number of responders: 2
Flow        State    Dir       Frm count
TCP           1.1.79.2:14083 ->        2.2.2.2:21    Watch    I              13
    NAT source        1.1.79.2:14083   ->   194.250.1.237:50118
TCP           1.1.79.2:14104 ->        2.2.2.2:20    Forward  I               3
    NAT source        1.1.79.2:14104   ->   194.250.1.237:50119
TCP            2.2.2.2:21    ->  194.250.1.237:50118 Watch    O              12
    NAT dest     194.250.1.237:50118   ->        1.1.79.2:14083
TCP            2.2.2.2:20    ->  194.250.1.237:50119 Forward  O               5
    NAT dest     194.250.1.237:50119   ->        1.1.79.2:14104
```

For each flow, the first line shows flow information, including protocol (TCP), source address, source port, destination address, destination port, flow state, direction, and frame count.

- The state of a flow can be `Watch`, `Forward`, or `Drop`:
  - A `Watch` flow state indicates that the control flow is monitored by the ALG for information in the payload. NAT processing is performed on the header and payload as needed.
  - A `Forward` flow forwards the packets without monitoring the payload. NAT is performed on the header as needed.
  - A `Drop` flow drops any packet that matches the 5 tuple.
- The frame count (`Frm count`) shows the number of packets that were processed on that flow.

The second line shows the NAT information.

- `source` indicates source NAT.

- `dest` indicates destination NAT.
- The first address and port in the NAT line are the original address and port being translated for that flow.
- The second address and port in the NAT line are the translated address and port for that flow.

MX-SPC3 Card

On the MX-SPC3 services card, the following is a complete sample output from the `show services sessions application-protocol ftp` operational mode command:

```
user@host>show services sessions application-protocol ftp
Session ID: 536870917, Service-set: ss1, Policy name: p1/131085, Timeout: 1, Valid
Logical system: root-logical-system
Resource information : FTP ALG, 1, 1
  In: 12.10.10.10/35281 --> 22.20.20.3/8204;tcp, Conn Tag: 0x0, If: vms-2/0/0.100, Pkts: 6,
Bytes: 320,
  Out: 22.20.20.3/8204 --> 60.1.1.2/48747;tcp, Conn Tag: 0x0, If: vms-2/0/0.200, Pkts: 9, Bytes:
8239,

Session ID: 536870919, Service-set: ss1, Policy name: p1/131085, Timeout: 29, Valid
Logical system: root-logical-system
Resource information : FTP ALG, 1, 0
  In: 12.10.10.10/44194 --> 22.20.20.3/21;tcp, Conn Tag: 0x0, If: vms-2/0/0.100, Pkts: 13,
Bytes: 585,
  Out: 22.20.20.3/21 --> 60.1.1.2/48660;tcp, Conn Tag: 0x0, If: vms-2/0/0.200, Pkts: 11, Bytes:
650,
Total sessions: 2
```

For each session:

- The first line shows flow information, including session ID, service-set name, policy name, session timeout, logical system name, and its state.
- The second line, `Resource information`, indicates the session is created by ALG, including the ALG name (FTP ALG) and ASL group id, which is 1and the ASL resource id, which is 0 for control session and 1 for data session.
- The third line `In` is forward flow and the fourth line `Out` is reverse flow, including the source address, source port, destination address, destination port, protocol (TCP), session conn-tag, incoming for

`In`and outgoing for `Out` interface, received frame count and bytes. NAT is performed on the header as needed.

FTP System Log Messages

System log messages are generated during an FTP session. For more information about system logs, see "System Log Messages" on page 543 .

MS-MPC Card

The following system log messages are generated during creation of the FTP control flow:

- Rule Accept system log:

```
Oct 27 11:42:54 (FPC Slot 1, PIC Slot 1) {ss_ftp}[FWNAT]: ASP_SFW_RULE_ACCEPT: proto 6 (TCP)
application: ftp, fe-3/3/3.0:1.1.1.2:4450 -> 2.2.2.2:21, Match SFW accept rule-set:, rule:
ftp, term: 1
```

- Create Accept Flow system log:

```
Oct 27 11:42:54 (FPC Slot 1, PIC Slot 1) {ss_ftp}[FWNAT]: ASP_SFW_CREATE_ACCEPT_FLOW: proto 6
(TCP) application: ftp, fe-3/3/3.0:1.1.1.2:4450 -> 2.2.2.2:21, creating forward or watch flow
```

- System log for data flow creation:

```
Oct 27 11:43:30 (FPC Slot 1, PIC Slot 1) {ss_ftp}[FWNAT]: ASP_SFW_FTP_ACTIVE_ACCEPT: proto 6
(TCP) application: ftp, so-2/1/2.0:2.2.2.2:20 -> 1.1.1.2:50726, Creating FTP active mode
forward flow
```

MX-SPC3 CardCard

The following system log messages are generated during creation of the FTP control flow:

- System log for FTP control session creation:

```
Mar 23 23:58:54  esst480r RT_FLOW: RT_FLOW_SESSION_CREATE_USF: Tag  svc-set-name ss1: session
created 20.1.1.2/52877->30.1.1.2/21 0x0 junos-ftp 20.1.1.2/52877->30.1.1.2/21 0x0 N/A N/A N/A
N/A 6 p1 ss1-ZoneIn ss1-ZoneOut 818413576 N/A(N/A) ge-1/0/2.0 UNKNOWN UNKNOWN UNKNOWN N/A N/A
-1 N/A
```

```
Mar 23 23:59:00  esst480r junos-alg: RT_ALG_FTP_ACTIVE_ACCEPT: application:ftp data,
vms-3/0/0.0 30.1.1.2:20  ->  20.1.1.2:33947 (TCP)
```

- System log for FTP data session creation:

```
Mar 23 23:59:00  esst480r RT_FLOW: RT_FLOW_SESSION_CREATE_USF: Tag  svc-set-name ss1: session
created 30.1.1.2/20->20.1.1.2/33947 0x0 junos-ftp-data 30.1.1.2/20->20.1.1.2/33947 0x0 N/A
N/A N/A N/A 6 p1 ss1-ZoneOut ss1-ZoneIn 818413577 N/A(N/A) ge-1/1/6.0 FTP-DATA UNKNOWN
UNKNOWN Infrastructure File-Servers 2 N/A
```

- System log for FTP data session destroy:

```
Mar 23 23:59:02  esst480r RT_FLOW: RT_FLOW_SESSION_CLOSE_USF: Tag  svc-set-name ss1: session
closed TCP FIN: 30.1.1.2/20->20.1.1.2/33947 0x0 junos-ftp-data 30.1.1.2/20->20.1.1.2/33947
0x0 N/A N/A N/A N/A 6 p1 ss1-ZoneOut ss1-ZoneIn 818413577 2954(4423509) 281(14620) 2 FTP-DATA
UNKNOWN N/A(N/A) ge-1/1/6.0 No Infrastructure File-Servers 2 N/A
```

- System log for FTP control session destroy:

```
Mar 23 23:59:39  esst480r RT_FLOW: RT_FLOW_SESSION_CLOSE_USF: Tag  svc-set-name ss1: session
closed Closed by junos-tcp-clt-emul: 20.1.1.2/52877->30.1.1.2/21 0x0 junos-ftp 20.1.1.2/52877-
>30.1.1.2/21 0x0 N/A N/A N/A N/A 6 p1 ss1-ZoneIn ss1-ZoneOut 818413576 23(1082) 18(1176) 45
UNKNOWN UNKNOWN N/A(N/A) ge-1/0/2.0 No N/A N/A -1 N/A
```

Analysis

Control Flows

MS-MPC Card

The control flows are established after the three-way handshake is complete.

- Control flow from FTP client to FTP server. TCP destination port is 21.

```
TCP           1.1.79.2:14083 ->        2.2.2.2:21    Watch    I              13
    NAT source        1.1.79.2:14083   ->   194.250.1.237:50118
```

- Control flow from FTP server to FTP client. TCP source port is 21.

```
TCP            2.2.2.2:21    ->  194.250.1.237:50118 Watch    O              12
    NAT dest     194.250.1.237:50118   ->        1.1.79.2:14083
```

MX-SPC3 Card

The control flows are established after the three-way handshake is complete.

- Control session from FTP client to FTP server, TCP destination port is 21.

```
Session ID: 536870919, Service-set: ss1, Policy name: p1/131085, Timeout: 29, Valid
Logical system: root-logical-system
Resource information : FTP ALG, 1, 0
  In: 12.10.10.10/44194 --> 22.20.20.3/21;tcp, Conn Tag: 0x0, If: vms-2/0/0.100, Pkts: 13,
Bytes: 585,
  Out: 22.20.20.3/21 --> 60.1.1.2/48660;tcp, Conn Tag: 0x0, If: vms-2/0/0.200, Pkts: 11,
Bytes: 650,
```

- Data session from FTP client to FTP server, it’s for FTP passive mode.

```
Session ID: 536870917, Service-set: ss1, Policy name: p1/131085, Timeout: 1, Valid
Logical system: root-logical-system
Resource information : FTP ALG, 1, 1
  In: 12.10.10.10/35281 --> 22.20.20.3/8204;tcp, Conn Tag: 0x0, If: vms-2/0/0.100, Pkts: 6,
Bytes: 320,
  Out: 22.20.20.3/8204 --> 60.1.1.2/48747;tcp, Conn Tag: 0x0, If: vms-2/0/0.200, Pkts: 9,
Bytes: 8239,
```

- Data session from FTP server to FTP client, it’s for FTP active mode:

```
Session ID: 549978117, Service-set: ss1, Policy name: p1/131085, Timeout: 1, Valid
Logical system: root-logical-system
```

```
Resource information : FTP ALG, 1, 1
  In: 22.20.20.3/20 --> 60.1.1.3/6049;tcp, Conn Tag: 0x0, If: vms-2/0/0.200, Pkts: 10, Bytes:
8291,
  Out: 12.10.10.10/33203 --> 22.20.20.3/20;tcp, Conn Tag: 0x0, If: vms-2/0/0.100, Pkts: 5,
Bytes: 268,
```

Data Flows

A data port of 20 is negotiated for data transfer during the course of the FTP control protocol. These two flows are data flows between the FTP client and the FTP server:

```
TCP           1.1.79.2:14104 ->        2.2.2.2:20    Forward  I               3
    NAT source        1.1.79.2:14104   ->   194.250.1.237:50119
TCP            2.2.2.2:20    ->  194.250.1.237:50119 Forward  O               5
    NAT dest     194.250.1.237:50119   ->        1.1.79.2:14104
```

Troubleshooting Questions

1. How do I know if the FTP ALG is active?

   - The ALG protocol field in the conversation should display `ftp`.
   - There should be a valid frame count (`Frm count`) in the control flows.
   - A valid frame count in the data flows indicates that data transfer has taken place.

2. What do I need to check if the FTP connection is established but data transfer does not take place?

   - Most probably, the control connection is up, but the data connection is down.
   - Check the conversations output to determine whether both the control and data flows are present.

3. How do I interpret each flow? What does each flow mean?

   - FTP control flow initiator flow—Flow with destination port 21
   - FTP control flow responder flow—Flow with source port ;21
   - FTP data flow initiator flow—Flow with destination port 20
   - FTP data flow responder flow—Flow with source port 20

RTSP ALG Example

The following is an example of an RTSP conversation. The application uses the RTSP protocol for control connection. Once the connection is set up, the media is sent using UDP protocol (RTP).

This example consists of the following:

Sample Output for MS-MPCs

Here is the output from the `show services stateful-firewall conversations` operational mode command:

```
user@host# show services stateful-firewall conversations
Interface: ms-3/2/0, Service set: svc_set
Conversation: ALG protocol: rtsp
  Number of initiators: 5, Number of responders: 5
Flow           State         Dir       Frm count
TCP            1.1.1.3:58795 ->        2.2.2.2:554   Watch    I      7
UDP            1.1.1.3:1028  ->        2.2.2.2:1028  Forward  I      0
UDP            1.1.1.3:1029  ->        2.2.2.2:1029  Forward  I      0
UDP            1.1.1.3:1030  ->        2.2.2.2:1030  Forward  I      0
UDP            1.1.1.3:1031  ->        2.2.2.2:1031  Forward  I      0
TCP            2.2.2.2:554   ->        1.1.1.3:58795 Watch    O      5
UDP            2.2.2.2:1028  ->        1.1.1.3:1028  Forward  O      6
UDP            2.2.2.2:1029  ->        1.1.1.3:1029  Forward  O      0
UDP            2.2.2.2:1030  ->        1.1.1.3:1030  Forward  O      3
UDP            2.2.2.2:1031  ->        1.1.1.3:1031  Forward  O      0
```

Sample Output for MX-SPC3 Services Card

Here is the output from the `show services sessions application-protocol rtsp` operational mode command:

```
user@host# run show services sessions application-protocol rtsp
Session ID: 1073741828, Service-set: sset1, Policy name: p1/131081, Timeout: 116, Valid
Logical system: root-logical-system
Resource information : RTSP ALG, 1, 0
  In: 31.0.0.2/33575 --> 41.0.0.2/554;tcp, Conn Tag: 0x0, If: vms-4/0/0.1, Pkts: 8, Bytes: 948,
  Out: 41.0.0.2/554 --> 131.10.0.1/7777;tcp, Conn Tag: 0x0, If: vms-4/0/0.2, Pkts: 6, Bytes:
1117,

Session ID: 1073741829, Service-set: sset1, Policy name: p1/131081, Timeout: 120, Valid
Logical system: root-logical-system
```

```
Resource information : RTSP ALG, 1, 1
  In: 41.0.0.2/35004 --> 131.10.0.1/7780;udp, Conn Tag: 0x0, If: vms-4/0/0.2, Pkts: 220, Bytes: 
79200,
  Out: 31.0.0.2/30004 --> 41.0.0.2/35004;udp, Conn Tag: 0x0, If: vms-4/0/0.1, Pkts: 0, Bytes: 0,

Session ID: 1073741830, Service-set: sset1, Policy name: p1/131081, Timeout: 120, Valid
Logical system: root-logical-system
Resource information : RTSP ALG, 1, 4
  In: 41.0.0.2/35006 --> 131.10.0.1/7781;udp, Conn Tag: 0x0, If: vms-4/0/0.2, Pkts: 220, Bytes: 
174240,
  Out: 31.0.0.2/30006 --> 41.0.0.2/35006;udp, Conn Tag: 0x0, If: vms-4/0/0.1, Pkts: 0, Bytes: 0,
Total sessions: 3
```

Analysis

An RTSP conversation should consist of TCP flows corresponding to the RTSP control connection. There should be two flows, one in each direction, from client to server and from server to client:

```
TCP              1.1.1.3:58795 ->        2.2.2.2:554   Watch    I               7
TCP              2.2.2.2:554   ->        1.1.1.3:58795 Watch    O               5
```

- The RTSP control connection for the initiator flow is sent from destination port 554.
- The RTSP control connection for the responder flow is sent from source port 554.

The UDP flows correspond to RTP media sent over the RTSP connection.

Troubleshooting Questions

1. Media does not work when the RTSP ALG is configured. What do I do?
   - Check RTSP conversations to see whether both TCP and UDP flows exist.
   - The ALG protocol should be displayed as `rtsp`.

   NOTE: The state of the flow is displayed as `Watch`, because the ALG processing is taking place and the client is essentially “watching” or processing payload corresponding to the application. For FTP and RTSP ALG flows, the control connections are always `Watch` flows.

2. How do I check for ALG errors?

- You can check for errors by issuing the following command. Each ALG has a separate field for ALG packet errors.

```
user@host# show services stateful-firewall statistics extensive
Interface: ms-3/2/0
  Service set: svc_set
    New flows:
      Accepts: 1347, Discards: 0, Rejects: 0
    Existing flows:
      Accepts: 144187, Discards: 0, Rejects: 0
    Drops:
      IP option: 0, TCP SYN defense: 0
      NAT ports exhausted: 0
    Errors:
      IP: 0, TCP: 276
      UDP: 0, ICMP: 0
      Non-IP packets: 0, ALG: 0
    IP errors:
      IP packet length inconsistencies: 0
      Minimum IP header length check failures: 0
      Reassembled packet exceeds maximum IP length: 0
      Illegal source address: 0
      Illegal destination address: 0
      TTL zero errors: 0, Illegal IP protocol number (0 or 255): 0
      Land attack: 0
      Non-IPv4 packets: 0, Bad checksum: 0
      Illegal IP fragment length: 0
      IP fragment overlap: 0
      IP fragment reassembly timeout: 0
      Unknown: 0
    TCP errors:
      TCP header length inconsistencies: 0
      Source or destination port number is zero: 0
      Illegal sequence number and flags combinations: 0
      SYN attack (multiple SYN messages seen for the same flow): 276
      First packet not a SYN message: 0
      TCP port scan (TCP handshake, RST seen from server for SYN): 0
      Bad SYN cookie response: 0
    UDP errors:
      IP data length less than minimum UDP header length (8 bytes): 0
      Source or destination port number is zero: 0
      UDP port scan (ICMP error seen for UDP flow): 0
    ICMP errors:
      IP data length less than minimum ICMP header length (8 bytes): 0
      ICMP error length inconsistencies: 0
      Duplicate ping sequence number: 0
      Mismatched ping sequence number: 0
    ALG errors:
      BOOTP: 0, DCE-RPC: 0, DCE-RPC portmap: 0
      DNS: 0, Exec: 0, FTP: 0
      ICMP: 0
      Login: 0, NetBIOS: 0, NetShow: 0
      RPC: 0, RPC portmap: 0
      RTSP: 0, Shell: 0
      SNMP: 0, SQLNet: 0, TFTP: 0
      Traceroute: 0
```

System Log Messages

Enabling system log generation and checking the system log are also helpful for ALG flow analysis. This section contains the following:

System Log Configuration

You can configure the enabling of system log messages at a number of different levels in the Junos OS CLI. As shown in the following sample configurations, the choice of level depends on how specific you want the event logging to be and what options you want to include. For details on the configuration options, see the Junos OS Administration Library for Routing Devices (system level) or the Junos OS Services Interfaces Library for Routing Devices (all other levels).

1. At the topmost global level:

```
user@host# show system syslog
file messages {
    any any;
}
```

2. At the service set level:

```
user@host# show services service-set svc_set
syslog {
    host local {
        services any;
    }
}
stateful-firewall-rules allow_rtsp;
interface-service {
    service-interface ms-3/2/0;
}
```

3. At the service rule level:

```
user@host# show services stateful-firewall rule allow_rtsp
match-direction input-output;
term 0 {
    from {
        applications junos-rtsp;
    }
    then {
        accept;
        syslog;
    }
}
```

System Log Output

System log messages are generated during flow creation, as shown in the following examples:

The following system log message indicates that the ASP matched an accept rule:

```
Oct 25 16:11:37  (FPC Slot 3, PIC Slot 2) {svc_set}[FWNAT]: ASP_SFW_RULE_ACCEPT: proto 6 (TCP)
application: rtsp, ge-2/0/1.0:1.1.1.2:35595 -> 2.2.2.2:554, Match SFW accept rule-set: , rule:
allow_rtsp, term: 0
```

For a complete listing of system log messages, see the System Log Explorer.

6

PART

Access Security

Stateful Firewalls | 546

IDS on MS-DPC | 588

Network Attack Protection on MS-MPC and MS-MIC | 605

CHAPTER 26

Stateful Firewalls

IN THIS CHAPTER

- Stateful Firewalls | 546
- Monitoring Stateful Firewalls | 586

Stateful Firewalls

IN THIS SECTION

- Junos Network Secure Overview | 546
- Configuring Stateful Firewall Rules | 550
- Configuring Stateful Firewall Rule Sets | 556
- Examples: Configuring Stateful Firewall Rules | 557
- Example: BOOTP and Broadcast Addresses | 561
- Example: Configuring Layer 3 Services and the Services SDK on Two PICs | 562
- Example: Virtual Routing and Forwarding (VRF) and Service Configuration | 583

Junos Network Secure Overview

IN THIS SECTION

- Stateful Firewall Support for Application Protocols | 548
- Stateful Firewall Anomaly Checking | 548

Routers use firewalls to track and control the flow of traffic. Adaptive Services and MultiServices PICs employ a type of firewall called a . Contrasted with a firewall that inspects packets in isolation, a stateful firewall provides an extra layer of security by using state information derived from past communications and other applications to make dynamic control decisions for new communication attempts.

NOTE: On ACX Series routers, the stateful firewall configuration is supported only on the ACX500 indoor routers.

Stateful firewalls group relevant into . A flow is identified by the following five properties:

- Source address
- Source port
- Destination address
- Destination port
- Protocol

A typical Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) conversation consists of two flows: the initiation flow and the responder flow. However, some conversations, such as an FTP conversation, might consist of two control flows and many data flows.

Firewall rules govern whether the conversation is allowed to be established. If a conversation is allowed, all flows within the conversation are permitted, including flows that are created during the life cycle of the conversation.

You configure stateful firewalls using a powerful rule-driven conversation handling path. A consists of direction, source address, source port, destination address, destination port, IP protocol value, and application protocol or service. In addition to the specific values you configure, you can assign the value `any` to rule objects, addresses, or ports, which allows them to match any input value. Finally, you can optionally negate the rule objects, which negates the result of the type-specific match.

Firewall rules are directional. For each new conversation, the router software checks the initiation flow matching the direction specified by the rule.

Firewall rules are ordered. The software checks the rules in the order in which you include them in the configuration. The first time the firewall discovers a match, the router implements the action specified by that rule. Rules still unchecked are ignored.

NOTE: Starting in Junos OS Release 14.2, MS-MPC and MS-MIC interface cards support IPv6 traffic for Junos Network Secure Stateful Firewall.

For more information, see "Configuring Stateful Firewall Rules" on page 550 .

Stateful Firewall Support for Application Protocols

By inspecting the application protocol data, the AS or MultiServices PIC firewall can intelligently enforce security policies and allow only the minimal required packet traffic to flow through the firewall.

The firewall rules are configured in relation to an interface. By default, the stateful firewall allows all sessions initiated from the hosts behind the interface to pass through the router.

> **NOTE**: Stateful firewall ALGs are not supported on ACX500 routers.

Stateful Firewall Anomaly Checking

The stateful firewall recognizes the following events as anomalies and sends them to the IDS software for processing:

- IP anomalies:
  - IP version is not correct.
  - IP header length field is too small.
  - IP header length is set larger than the entire packet.
  - Bad header checksum.
  - IP total length field is shorter than header length.
  - Packet has incorrect IP options.
  - Internet Control Message Protocol (ICMP) packet length error.
  - Time-to-live (TTL) equals 0.
- IP address anomalies:
  - IP packet source is a broadcast or multicast.
  - Land attack (source IP equals destination IP).
- IP fragmentation anomalies:
  - IP fragment overlap.
  - IP fragment missed.

  - IP fragment length error.
  - IP packet length is more than 64 kilobytes (KB).
  - Tiny fragment attack.
- TCP anomalies:
  - TCP port 0.
  - TCP sequence number 0 and flags 0.
  - TCP sequence number 0 and FIN/PSH/RST flags set.
  - TCP flags with wrong combination (TCP FIN/RST or SYN/(URG|FIN|RST).
  - Bad TCP checksum.
- UDP anomalies:
  - UDP source or destination port 0.
  - UDP header length check failed.
  - Bad UDP checksum.
- Anomalies found through stateful TCP or UDP checks:
  - SYN followed by SYN-ACK packets without ACK from initiator.
  - SYN followed by RST packets.
  - SYN without SYN-ACK.
  - Non-SYN first flow packet.
  - ICMP unreachable errors for SYN packets.
  - ICMP unreachable errors for UDP packets.
- Packets dropped according to stateful firewall rules.

NOTE: ACX500 routers do not support IP fragmentation anomalies.

If you employ stateful anomaly detection in conjunction with stateless detection, IDS can provide early warning for a wide range of attacks, including these:

- TCP or UDP network probes and port scanning

- SYN flood attacks
- IP fragmentation-based attacks such as teardrop, bonk, and boink

Configuring Stateful Firewall Rules

IN THIS SECTION

- Configuring Match Direction for Stateful Firewall Rules | **552**
- Configuring Match Conditions in Stateful Firewall Rules | **553**
- Configuring Actions in Stateful Firewall Rules | **554**

To configure a stateful firewall rule, include the `rule` *`rule-name`* statement at the `[edit services stateful-firewall]` hierarchy level:

```
[edit services stateful-firewall]
rule rule-name {
    match-direction (input | output | input-output);
    term term-name {
        from {
            application-sets set-name;
            applications [ application-names ];
            destination-address  (address | any-ipv4 | any-ipv6 | any-unicast) <except>;
            destination-address-range low minimum-value high maximum-value <except>;
            destination-prefix-list list-name <except>;
            source-address  (address | any-ipv4 | any-ipv6 | any-unicast) <except>;
            source-address-range low minimum-value high maximum-value <except>;
            source-prefix-list list-name <except>;
        }
        then {
            (accept <skip-ids>| discard | reject);
            allow-ip-options [ values ];
            syslog;
        }
    }
}
```

NOTE: ACX500 routers do not support *applications* and *application-sets* at the [edit services stateful-firewall rule *rule-name* term *term-name* from] hierarchy level.

NOTE: On ACX500 routers, to enable syslog, include the stateful-firewall-logs CLI statement at the [edit services service-set service-set-name syslog host local class] hierarchy level.

NOTE: edit services stateful-firewall hierarchy is not supported on SRX series.

Each stateful firewall rule consists of a set of terms, similar to a filter configured at the [edit firewall] hierarchy level. A term consists of the following:

- from statement—Specifies the match conditions and applications that are included and excluded. The from statement is optional in stateful firewall rules.
- then statement—Specifies the actions and action modifiers to be performed by the router software. The then statement is mandatory in stateful firewall rules.

ACX500 Series routers do not support the following while configuring stateful firewall rules:

- match-direction (**output | input-output**)
- post-service-filter at the interface service input hierarchy level.
- IPv6 source address and destination address.
- application-sets, application, allow-ip-options at the [edit services stateful-firewall] hierarchy level.
- Application Layer Gateways (ALGs).
- Chaining of services within Multiservices Modular Interfaces Card (MS-MIC) and with inline-services (-si).
- Class of service.
- The following show services stateful-firewall CLI commands are not supported:
  - show services stateful-firewall conversations—Show conversations
  - show services stateful-firewall flow-analysis—Show flow table entries
  - show services stateful-firewall redundancy-statistics—Show redundancy statistics

- `show services stateful-firewall sip-call`—Show SIP call information
- `show services stateful-firewall sip-register`—Show SIP register information
- `show services stateful-firewall subscriber-analysis`—Show subscriber table entries

The following sections explain how to configure the components of stateful firewall rules:

Configuring Match Direction for Stateful Firewall Rules

Each rule must include a `match-direction` statement that specifies the direction in which the rule match is applied. To configure where the match is applied, include the `match-direction` statement at the `[edit services stateful-firewall rule` *`rule-name`*`]` hierarchy level:

```
[edit services stateful-firewall rule rule-name]
match-direction (input | output | input-output);
```

NOTE: ACX500 Series routers do not support `match-direction (output | input-output)`.

If you configure `match-direction input-output`, sessions initiated from both directions might match this rule.

The match direction is used with respect to the traffic flow through the AS or Multiservices PIC. When a packet is sent to the PIC, direction information is carried along with it.

With an interface service set, packet direction is determined by whether a packet is entering or leaving the interface on which the service set is applied.

With a next-hop service set, packet direction is determined by the interface used to route the packet to the AS or Multiservices PIC. If the inside interface is used to route the packet, the packet direction is input. If the outside interface is used to direct the packet to the PIC, the packet direction is output. For more information on inside and outside interfaces, see "Configuring Service Sets to be Applied to Services Interfaces" on page 10 .

On the PIC, a flow lookup is performed. If no flow is found, rule processing is performed. Rules in this service set are considered in sequence until a match is found. During rule processing, the packet direction is compared against rule directions. Only rules with direction information that matches the packet direction are considered. Most packets result in the creation of bidirectional flows.

Configuring Match Conditions in Stateful Firewall Rules

To configure stateful firewall match conditions, include the `from` statement at the `[edit services stateful-firewall rule rule-name term term-name]` hierarchy level:

```
[edit services stateful-firewall rule rule-name term term-name]
from {
    application-sets set-name;
    applications [ application-names ];
    destination-address (address | any-ipv4 | any-ipv6 | any-unicast) <except>;
    destination-address-range low minimum-value high maximum-value <except>;
    destination-prefix-list list-name <except>;
    source-address (address | any-ipv4 | any-ipv6 | any-unicast) <except>;
    source-address-range low minimum-value high maximum-value <except>;
    source-prefix-list list-name <except>;
}
```

NOTE: ACX500 routers do not support *applications* and *application-sets* at the `[edit services stateful-firewall rule rule-name term term-name from]` hierarchy level.

The source address and destination address can be either IPv4 or IPv6.

You can use either the source address or the destination address as a match condition, in the same way that you would configure a firewall filter; for more information, see the Routing Policies, Firewall Filters, and Traffic Policers User Guide. You can use the wildcard values `any-unicast`, which denotes matching all unicast addresses, `any-ipv4`, which denotes matching all IPv4 addresses, or `any-ipv6`, which denotes matching all IPv6 addresses.

Alternatively, you can specify a list of source or destination prefixes by configuring the `prefix-list` statement at the `[edit policy-options]` hierarchy level and then including either the `destination-prefix-list` or the `source-prefix-list` statement in the stateful firewall rule. For an example, see "Examples: Configuring Stateful Firewall Rules" on page 557 .

If you omit the `from` term, the stateful firewall accepts all traffic and the default protocol handlers take effect:

- User Datagram Protocol (UDP), Transmission Control Protocol (TCP), and Internet Control Message Protocol (ICMP) create a bidirectional flow with a predicted reverse flow.

- IP creates a unidirectional flow.

You can also include application protocol definitions you have configured at the `[edit applications]` hierarchy level; for more information, see "Configuring Application Properties" on page 508 .

- To apply one or more specific application protocol definitions, include the `applications` statement at the `[edit services stateful-firewall rule rule-name term term-name from]` hierarchy level.
- To apply one or more sets of application protocol definitions you have defined, include the `application-sets` statement at the `[edit services stateful-firewall rule rule-name term term-name from]` hierarchy level.

> **NOTE**: If you include one of the statements that specifies application protocols, the router derives port and protocol information from the corresponding configuration at the `[edit applications]` hierarchy level; you cannot specify these properties as match conditions.

Configuring Actions in Stateful Firewall Rules

To configure stateful firewall actions, include the `then` statement at the `[edit services stateful-firewall rule rule-name term term-name]` hierarchy level:

```
[edit services stateful-firewall rule rule-name term term-name]
then {
    (accept | discard | reject);
    allow-ip-options [ values ];
    syslog;
}
```

You must include one of the following actions:

- `accept`—The packet is accepted and sent on to its destination.
- `accept skip-ids`—The packet is accepted and sent on to its destination, but IDS rule processing configured on an MS-MPC is skipped.
- `discard`—The packet is not accepted and is not processed further.
- `reject`—The packet is not accepted and a rejection message is returned; UDP sends an ICMP unreachable code and TCP sends RST. Rejected packets can be logged or sampled.

> **NOTE**: The ACX500 indoor routers do not support the action `accept skip-ids`.

You can optionally configure the firewall to record information in the system logging facility by including the syslog statement at the [edit services stateful-firewall rule *rule-name* term *term-name* then] hierarchy level. This statement overrides any syslog setting included in the service set or interface default configuration.

Configuring IP Option Handling

You can optionally configure the firewall to inspect IP header information by including the allow-ip-options statement at the [edit services stateful-firewall rule *rule-name* term *term-name* then] hierarchy level. When you configure this statement, all packets that match the criteria specified in the from statement are subjected to additional matching criteria. A packet is accepted only when all of its IP option types are configured as values in the allow-ip-options statement. If you do not configure allow-ip-options, only packets without IP header options are accepted.

NOTE: ACX500 indoor routers do not support the configuration of allow-ip-options statement.

The additional IP header option inspection applies only to the accept and reject stateful firewall actions. This configuration has no effect on the discard action. When the IP header inspection fails, reject frames are not sent; in this case, the reject action has the same effect as discard.

If an IP option packet is accepted by the stateful firewall, Network Address Translation (NAT) and intrusion detection service (IDS) are applied in the same way as to packets without IP option headers. The IP option configuration appears only in the stateful firewall rules; NAT applies to packets with or without IP options.

When a packet is dropped because it fails the IP option inspection, this exception event generates both IDS event and system log messages. The event type depends on the first IP option field rejected.

Table 23 on page 555 lists the possible values for the allow-ip-options statement. You can include a range or set of numeric values, or one or more of the predefined IP option settings. You can enter either the option name or its numeric equivalent. For more information, refer to http://www.iana.org/assignments/ip-parameters.

Table 23: IP Option Values

IP Option Name	Numeric Value	Comment
any	0	Any IP option
ip-security	130	–

Table 23: IP Option Values *(Continued)*

IP Option Name	Numeric Value	Comment
ip-stream	136	–
loose-source-route	131	–
route-record	7	–
router-alert	148	–
strict-source-route	137	–
timestamp	68	–

SEE ALSO

Junos Network Secure Overview | 546

Configuring Protection Against Network Attacks on an MS-MPC | 610

Configuring Stateful Firewall Rule Sets

The `rule-set` statement defines a collection of stateful firewall rules that determine what actions the router software performs on packets in the data stream. You define each rule by specifying a rule name and configuring terms. Then, you specify the order of the rules by including the `rule-set` statement at the `[edit services stateful-firewall]` hierarchy level with a `rule` statement for each rule:

```
[edit services stateful-firewall]
rule-set rule-set-name {
    rule rule-name;
}
```

The router software processes the rules in the order in which you specify them in the configuration. If a term in a rule matches the packet, the router performs the corresponding action and the rule processing stops. If no term in a rule matches the packet, processing continues to the next rule in the rule set. If none of the rules matches the packet, the packet is dropped by default.

Examples: Configuring Stateful Firewall Rules

The following example shows a stateful firewall configuration containing two rules, one for input matching on a specified application set and the other for output matching on a specified source address:

```
[edit services]
stateful-firewall {
    rule Rule1 {
        match-direction input;
        term 1 {
            from {
                application-sets Applications;
            }
            then {
                accept;
            }
        }
        term accept {
            then {
                accept;
            }
        }
    }
    rule Rule2 {
        match-direction output;
        term Local {
            from {
                source-address {
                    10.1.3.2/32;
                }
            }
            then {
                accept;
            }
        }
    }
}
```

The following example has a single rule with two terms. The first term rejects all traffic in `my-application-group` that originates from the specified source address, and provides a detailed system log record of the rejected packets. The second term accepts Hypertext Transfer Protocol (HTTP) traffic from anyone to the specified destination address.

```
[edit services stateful-firewall]
rule my-firewall-rule {
    match-direction input-output;
    term term1 {
        from {
            source-address 10.1.3.2/32;
            application-sets my-application-group;
        }
        then {
            reject;
            syslog;
        }
    }
    term term2 {
        from {
            destination-address 10.2.3.2/32;
            applications http;
        }
        then {
            accept;
        }
    }
}
```

The following example shows use of source and destination prefix lists. This requires two separate configuration items.

You configure the prefix list at the `[edit policy-options]` hierarchy level:

```
[edit]
policy-options {
    prefix-list p1 {
        10.1.1.1/32;
        10.2.2.0/24;
    }
    prefix-list p2 {
        10.3.3.3/32;
        10.4.4.0/24;
    }
}
```

You reference the configured prefix list in the stateful firewall rule:

```
[edit]
services {
    stateful-firewall {
        rule r1 {
            match-direction input;
            term t1 {
                from {
                    source-prefix-list {
                        p1;
                    }
                    destination-prefix-list {
                        p2;
                    }
                }
                then {
                    accept;
                }
            }
        }
    }
}
```

This is equivalent to the following configuration:

```
[edit]
services {
    stateful-firewall {
        rule r1 {
            match-direction input;
            term t1 {
                from {
                    source-address {
                        10.1.1.1/32;
                        10.2.2.0/24;
                    }
                    destination-address {
                        10.3.3.3/32;
                        10.4.4.0/24;
                    }
                }
                then {
                    accept;
                }
            }
        }
    }
}
```

You can use the `except` qualifier with the prefix lists, as in the following example. In this case, the `except` qualifier applies to all prefixes included in prefix list `p2`.

```
[edit]
services {
    stateful-firewall {
        rule r1 {
            match-direction input;
            term t1 {
                from {
                    source-prefix-list {
                        p1;
                    }
                    destination-prefix-list {
                        p2 except;
                    }
                }
                then {
                    accept;
                }
            }
        }
    }
}
```

For additional examples that combine stateful firewall configuration with other services and with virtual private network (VPN) routing and forwarding (VRF) tables, see the configuration examples.

NOTE: You can define the service-set and assign it either as interface style or next-hop style.

SEE ALSO

- Example: Dynamic Source NAT as a Next-Hop Service | **294**
- Example: Virtual Routing and Forwarding (VRF) and Service Configuration | **583**
- *Example: Service Interfaces Configuration*
- Configuring Service Sets to be Applied to Services Interfaces | **10**
- Example: Configuring Layer 3 Services and the Services SDK on Two PICs | **562**

Example: BOOTP and Broadcast Addresses

The following example supports Bootstrap Protocol (BOOTP) and broadcast addresses:

```
[edit applications]
application bootp {
    application-protocol bootp;
    protocol udp;
    destination-port 67;
}
[edit services]
stateful-firewall bootp-support {
    rule bootp-allow {
        direction input;
        term bootp-allow {
            from {
                destination-address {
                    any-unicast;
                    255.255.255.255;
                }
                application bootp;
            }
            then {
                accept;
            }
        }
    }
}
```

Example: Configuring Layer 3 Services and the Services SDK on Two PICs

You can configure the Layer 3 service package and the Services SDK on two PICs. For this example, you must configure an FTP or HTTP client and a server. In this configuration, the client side of the router interface is ge-1/2/2.1 and the server side of the router interface is ge-1/1/0.48. This configuration enables Network Address Translation (NAT) with stateful firewall (SFW) on the uKernel PIC and application identification (APPID), application-aware access list (AACL), and intrusion detection and prevention (IDP) on the Services SDK PIC for FTP or HTTP traffic.

NOTE: The Services SDK does not support NAT yet. When NAT is required, you can configure the Layer 3 service package to deploy NAT along with the Services SDK such as APPID, AACL, or IDP.

NOTE: The IDP functionality is deprecated for the MX Series for Junos OS release 17.1R1 and above.

To deploy the Layer 3 service package and the Services SDK on two PICs:

1. In configuration mode, go to the following hierarchy level:

   ```
   [edit services]
   user@host# edit stateful-firewall
   ```

2. In the hierarchy level, configure the conditions for the stateful firewall rule **r1**.

   ```
   [edit services stateful-firewall]
   user@host# set rule rule-name match-direction input-output term term from applications
   application-name
   user@host# set rule rule-name match-direction input-output term term then accept syslog
   ```

In this example, the stateful firewall term is **ALLOWED-SERVICES**. Enclose the application names—junos-ftp, junos-http, and junos-icmp-ping—in brackets for *application-name*.

```
[edit services stateful-firewall]
user@host# set rule r1 match-direction input-output term ALLOWED-SERVICES from applications
[ junos-ftp junos-http junos-icmp-ping ]
user@host# set rule r1 match-direction input-output term ALLOWED-SERVICES then accept syslog
```

3. Configure the conditions for the stateful firewall rule **r2**.

```
[edit services stateful-firewall]
user@host# set rule rule-name match-direction input-output term term then discard
user@host# set rule rule-name match-direction input-output term term then syslog
```

In this example, the stateful firewall term is **term1**.

```
[edit services stateful-firewall]
user@host# set rule r2 match-direction input-output term term1 then discard
user@host# set rule r2 match-direction input-output term term1 then syslog
```

4. Go to the following hierarchy level and verify the configuration:

```
[edit services stateful-firewall]
user@host# show
rule r1 {
    match-direction input-output;
    term ALLOWED-SERVICES {
        from {
            applications [ junos-ftp junos-http junos-icmp-ping ];
        }
        then {
            accept;
            syslog;
        }
    }
}
```

```
rule r2 {
    match-direction input-output;
    term term1 {
        then {
            discard;
            syslog;
        }
    }
}
```

5. Go to the following hierarchy level:

```
[edit services]
user@host# edit nat
```

6. In the hierarchy level, configure the NAT pool.

```
[edit services nat]
user@host# set pool pool-name address ip-address
user@host# set pool pool-name port automatic
```

In this example, the NAT pool is **OUTBOUND-SERVICES** and the IP address is **10.48.0.2/32**.

```
[edit services natl]
user@host# set pool OUTBOUND-SERVICES address 10.48.0.2/32
user@host# set pool OUTBOUND-SERVICES port automatic
```

7. Configure the NAT rule.

```
[edit services nat]
user@host# set rule rule-name match-direction output term term from applications
application-name
user@host# set rule rule-name match-direction output term term then translated source-pool
source-pool translation-type source dynamic
```

In this example, the NAT rule is **SET-MSR-ADDR**, the NAT term is **TRANSLATE-SOURCE-ADDR**, and the source pool is **OUTBOUND-SERVICES**. Enclose the application names—junos-ftp, junos-http, and junos-icmp-ping—in parentheses for *application-name*.

```
[edit services nat]
user@host# set rule SET-MSR-ADDR match-direction output term TRANSLATE-SOURCE-ADDR from 
applications [ junos-ftp junos-http junos-icmp-ping ]
user@host# set rule SET-MSR-ADDR match-direction output term TRANSLATE-SOURCE-ADDR then 
translated source-pool OUTBOUND-SERVICES translation-type source dynamic
```

8. Go to the following hierarchy level and verify the configuration:

```
[edit services nat]
user@host# show
pool OUTBOUND-SERVICES {
    address 11.48.0.2/32;
    port {
        automatic;
    }
}
rule SET-MSR-ADDR {
    match-direction output;
    term TRANSLATE-SOURCE-ADDR {
        from {
            applications [ junos-ftp junos-http junos-icmp-ping ];
        }
        then {
            translated {
                source-pool OUTBOUND-SERVICES;
                translation-type {
                    source dynamic;
                }
            }
        }
    }
}
```

9. Go to the following hierarchy level:

```
[edit security]
user@host# edit idp
```

NOTE: The [edit security idp] statements are deprecated for the MX Series for Junos OS release 17.1R1 and above.

10. In the hierarchy level, configure the IDP policy.

```
[edit security idp]
user@host# set idp-policy policy-name rulebase-ips rule rule-name match application default
attacks predefined-attacks attack-name
user@host# set idp-policy policy-name rulebase-ips rule rule-name match application default
attacks predefined-attack-groups attack-group--name
user@host# set idp-policy policy-name rulebase-ips rule rule-name then action no-action
user@host# set idp-policy policy-name rulebase-ips rule rule-name then notification log-
attacks alert
```

In this example, the IDP policy is **test1**, the rule is **r1**, the predefined attack is **FTP:USER:ROOT**, and the predefined attack group is **"Recommended Attacks"**.

```
[edit security idp]
user@host# set idp-policy test1 rulebase-ips rule r1 match application default attacks
predefined-attacks FTP:USER:ROOT
user@host# set idp-policy test1 rulebase-ips rule r1 match application default attacks
predefined-attack-groups [ "Recommended Attacks" ]
user@host# set idp-policy test1 rulebase-ips rule r1 then action no-action
user@host# set idp-policy test1 rulebase-ips rule r1 then notification log-attacks alert
```

11. Configure the trace options for IDP services.

```
[edit security idp]
user@host# set traceoptions file filename
user@host# set traceoptions flag all
user@host# set traceoptions level all
```

In this example, the log file name is **idp-demo.log**.

```
[edit security idp]
user@host# set traceoptions file idp-demo.log
user@host# set traceoptions flag all
user@host# set traceoptions level all
```

12. Go to the following hierarchy level and verify the configuration:

```
[edit security idp]
user@host# show
idp-policy test1 {
    rulebase-ips {
        rule r1 {
            match {
                application default;
                attacks {
                    predefined-attacks FTP:USER:ROOT;
                    predefined-attack-groups "Recommended Attacks";
                }
            }
            then {
                action {
                    no-action;
                }
                notification {
                    log-attacks {
                        alert;
                    }
                }
            }
        }
    }
}
traceoptions {
    file idp-demo.log;
    flag all;
    level all;
}
```

13. Go to the following hierarchy level:

    ```
    [edit services]
    user@host# edit aacl
    ```

14. In the hierarchy level, configure the AACL rules.

    ```
    [edit services aacl]
    user@host# set rule rule-name  match-direction input-output term term from application-group-any
    user@host# set rule rule-name  match-direction input-output term term then count application accept
    ```

    In this example, the AACL rule is **app-aware** and the term is **t1**.

    ```
    [edit services aacl]
    user@host# set rule app-aware  match-direction input-output term t1 from application-group-any
    user@host# set rule app-aware  match-direction input-output term t1 then count application accept
    ```

15. Go to the following hierarchy level and verify the configuration:

    ```
    [edit services aacl]
    user@host# show
    rule app-aware {
                match-direction input-output;
                term t1 {
                    from {
                        application-group-any;
                    }
                    then {
                        count application;
                        accept;
                    }
                }
            }
    ```

16. Go to the following hierarchy level:

```
[edit services]
user@host# edit service-set App-Aware-Set
```

17. Configure the APPID profile.

```
[edit services service-set App-Aware-Set]
user@host# set application-identification-profile application-identification-profile
```

In this example, the APPID profile is **dummy-profile**.

```
[edit services service-set App-Aware-Set]
user@host# set application-identification-profile dummy-profile
```

18. Configure the IDP profile.

```
[edit services service-set App-Aware-Set]
user@host# set idp-profile idp-profile
```

In this example, the IDP profile is **test1**.

```
[edit services service-set App-Aware-Set]
user@host# set idp-profile test1
```

19. Configure the policy decision statistics profile.

```
[edit services service-set App-Aware-Set]
user@host# set policy-decision-statistics-profile profile-name
```

In this example, the policy decision statistics profile is **lpdf-stats**.

```
[edit services service-set App-Aware-Set]
user@host# set policy-decision-statistics-profile lpdf-stats
```

20. Configure the AACL rules.

```
[edit services service-set App-Aware-Set]
user@host# set aacl-rules rule-name
```

In this example, the AACL rule name is **app-aware**.

```
[edit services service-set App-Aware-Set]
user@host# set aacl-rules app-aware
```

21. Configure two stateful firewall rules.

```
[edit services service-set App-Aware-Set]
user@host# set stateful-firewall-rules rule-name
user@host# set stateful-firewall-rules rule-name
```

In this example, the first rule is **r1** and the second rule is **r2**.

```
[edit services service-set App-Aware-Set]
user@host# set stateful-firewall-rules r1
user@host# set stateful-firewall-rules r2
```

22. In the hierarchy level, configure the service set to bypass traffic on service PIC failure.

```
[edit services service-set App-Aware-Set]
user@host# set service-set-options bypass-traffic-on-pic-failure
```

23. Configure interface-specific service set options.

```
[edit services service-set App-Aware-Set]
user@host# set interface-service service-interface service-interface
```

In this example, the services interface is **ms-0/1/0**.

```
[edit services service-set App-Aware-Set]
user@host# set interface-service service-interface ms-0/1/0
```

24. Go to the following hierarchy level and verify the configuration:

```
[edit services service-set App-Aware-Set]
user@host# show
application-identification-profile dummy-profile;
idp-profile test1;
policy-decision-statistics-profile {
    lpdf-stats;
}
aacl-rules app-aware;
stateful-firewall-rules r1;
stateful-firewall-rules r2;
service-set-options {
    bypass-traffic-on-pic-failure;
}
interface-service {
    service-interface ms-0/1/0;
}
```

25. Go to the following hierarchy level:

```
[edit services]
user@host# edit service-set NAT-SFW-SET
```

26. In the hierarchy level, configure optional notification parameters for the services interface. Note that it is required only for debugging.

```
[edit services service-set NAT-SFW-SET]
user@host# set syslog host host-name services any
```

In this example, the host to notify is **local**.

```
[edit services service-set NAT-SFW-SET]
user@host# set services-options syslog host local services any
```

27. Configure two stateful firewall rules.

```
[edit services service-set NAT-SFW-SET]
user@host# set stateful-firewall-rules rule-name
user@host# set stateful-firewall-rules rule-name
```

In this example, the first rule is **r1** and the second rule is **r2**.

```
[edit services service-set NAT-SFW-SET]
user@host# set stateful-firewall-rules r1
user@host# set stateful-firewall-rules r2
```

28. Configure NAT rules.

```
[edit services service-set  NAT-SFW-SET]
user@host# set nat-rules rule-name
```

In this example, the NAT rule is **SET-MSR-ADDR**.

```
[edit services service-set  NAT-SFW-SET]
user@host# set nat-rules SET-MSR-ADDR
```

29. Configure interface-specific service set options.

```
[edit services service-set NAT-SFW-SET]
user@host# set interface-service service-interface service-interface
```

In this example, the services interface is **sp-3/1/0**.

```
[edit services service-set NAT-SFW-SET]
user@host# set interface-service service-interface sp-3/1/0
```

30. Go to the following hierarchy level and verify the configuration:

```
[edit services service-set NAT-SFW-SET]
user@host# show
syslog {
    host local {
        services any;
    }
}
stateful-firewall-rules r1;
stateful-firewall-rules r2;
interface-service {
    service-interface sp-3/1/0;
}
```

31. Go to the following hierarchy level:

```
user@host# edit interfaces
```

32. In the hierarchy level, configure the interface.

```
 [edit interfaces]
user@host# set interface
```

In this example, the interface is **ge-1/2/2.1**.

```
[edit interfaces]
user@host# set ge-1/2/2.1
```

33. Go to the following hierarchy level:

```
 [edit interfaces]
user@host# edit ge-1/2/2.1
```

34. In the hierarchy level, configure the service set for received packets.

```
 [edit interfaces ge-1/2/2 unit 1]
user@host#  set family inet service input service-set service-set-name
```

In this example, the input service set is **App-Aware-Set**.

```
 [edit interfaces ge-1/2/2 unit 1]
user@host# set family inet service input service-set App-Aware-Set
```

35. Configure the service set for transmitted packets.

```
 [edit interfaces ge-1/2/2 unit 1]
user@host#  set family inet service output service-set service-set-name
```

In this example, the output service set is **App-Aware-Set**.

```
 [edit interfaces ge-1/2/2 unit 1]
user@host# set family inet service output service-set App-Aware-Set
```

36. Go to the following hierarchy level:

```
 [edit interfaces ge-1/2/2 unit 1]
user@host# edit family inet
```

37. In the hierarchy level, configure the interface address.

```
 [edit interfaces ge-1/2/2 unit 1 family inet]
user@host# set address source
```

In this example, the interface address is **10.10.9.10/30**.

```
 [edit interfaces]
user@host# set address 10.10.9.10/30
```

38. Go to the following hierarchy level and verify the configuration:

```
[edit interfaces ge-1/2/2 unit 1]
user@host# show
family inet {
    service {
        input {
            service-set App-Aware-Set;
        }
        output {
            service-set App-Aware-Set;
        }
    }
    address 10.10.9.10/30;
}
```

39. Go to the following hierarchy level:

```
user@host# edit interfaces
```

40. In the hierarchy level, configure the interface.

```
 [edit interfaces]
user@host# set interface
```

In this example, the interface is ge-1/1/0.48.

```
[edit interfaces]
user@host# set ge-1/1/0.48
```

41. Go to the following hierarchy level:

```
 [edit interfaces]
user@host# edit ge-1/1/0.48
```

42. In the hierarchy level, configure the service set for received packets.

```
[edit interfaces ge-1/1/0 unit 48]
user@host# set family inet service input service-set service-set-name
```

In this example, the service set is NAT-SFW-SET.

```
[edit interfaces ge-1/1/0 unit 48]
user@host# set family inet service input service-set NAT-SFW-SET
```

43. Configure the service set for transmitted packets.

```
[edit interfaces ge-1/1/0 unit 48]
user@host# set family inet service output service-set service-set-name
```

In this example, the service set is NAT-SFW-SET.

```
[edit interfaces ge-1/1/0 unit 48]
user@host# set family inet service output service-set NAT-SFW-SET
```

44. Go to the following hierarchy level:

```
[edit interfaces ge-1/1/0 unit 48]
user@host# edit family inet
```

45. Configure the interface address.

```
 [edit interfaces ge-1/1/0 unit 48 family inet]
user@host# set address source
```

In this example, the interface address is **10.48.0.1/31**.

```
[edit interfaces ge-1/1/0 unit 48 family inet]
user@host# set address 10.48.0.1/31
```

46. Go to the following hierarchy level and verify the configuration:

```
[edit interfaces ge-1/1/0 unit 48]
user@host# show
family inet {
    service {
        input {
            service-set NAT-SFW-SET;
            }
        output {
            service-set NAT-SFW-SET;
        }
    }
    address 10.48.0.1/31;
}
```

47. Go to the following hierarchy level:

```
user@host# edit interfaces
```

48. In the hierarchy level, configure the interface.

```
[edit interfaces]
set interface
```

In this example, the interface is **ms-0/1/0.0**.

```
[edit interfaces]
user@host# set ms-0/1/0.0
```

49. Go to the following hierarchy level:

```
[edit interfaces]
user@host# edit ms-0/1/0.0
```

50. In the hierarchy level, configure the protocol family.

```
[edit interfaces ms-0/1/0 unit 0]
user@host# set family inet
```

51. Go to the following hierarchy level and verify the configuration:

```
[edit interfaces ms-0/1/0]
user@host# show
unit 0 {
    family inet;
}
```

52. Go to the following hierarchy level:

```
user@host# edit interfaces
```

53. In the hierarchy level, configure the interface.

```
[edit interfaces]
set interface
```

In this example, the interface is **sp-3/1/0.0**.

```
[edit interfaces]
user@host# set sp-3/1/0.0
```

54. Go to the following hierarchy level:

```
[edit interfaces]
user@host# edit sp-3/1/0
```

55. In the hierarchy level, configure optional notification parameters for the services interface. Note that it is required only for debugging.

```
[edit interfaces sp-3/1/0]
user@host# set services-options syslog host host-name services any
```

In this example, the host to notify is **local**.

```
[edit interfaces sp-3/1/0]
user@host# set services-options syslog host local services any
```

56. Go to the following hierarchy level:

```
[edit interfaces]
user@host# edit sp-3/1/0.0
```

57. In the hierarchy level, configure the protocol family.

```
[edit interfaces sp-3/1/0 unit 0]
user@host# set family inet
```

58. Go to the following hierarchy level and verify the configuration:

```
[edit interfaces sp-3/1/0]
user@host# show
services-options {
    syslog {
        host local {
            services any;
        }
    }
}
unit 0 {
    family inet;
}
```

59. Go to the following hierarchy level:

```
[edit chassis]
```

60. In the hierarchy level, configure the redundancy settings.

```
[edit chassis]
user@host# set no-service-pic-restart-on-failover
user@host# set redundancy graceful-switchover
```

61. Configure the FPC and PIC.

```
 [edit chassis]
user@host# edit fpc slot pic slot
```

In this example, the FPC is in slot 0 and the PIC is in slot 1.

```
 [edit chassis]
user@host# edit fpc 0 pic 1
```

62. Configure the number of cores dedicated to run control functionality.

```
 [edit chassis fpc slot pic slot]
user@host#  set adaptive-services service-package extension-provider control-cores control-
cores
```

In this example, the number of control cores is 1.

```
 [edit chassis fpc 1 pic 0]
user@host#  set adaptive-services service-package extension-provider control-cores 1
```

63. Configure the number of processing cores dedicated to data.

```
 [edit chassis fpc slot pic slot]
user@host# set adaptive-services service-package extension-provider data-cores data-cores
```

In this example, the number of data cores is 7.

```
 [edit chassis fpc 1 pic 0]
user@host# set adaptive-services service-package extension-provider data-cores 7
```

64. Configure the size of the object cache in megabytes. Only values in increments of 128 MB are allowed and the maximum value of object cache can be 1280 MB. On MS-100, the value is 512 MB.

```
 [edit chassis fpc slot pic slot]
user@host# set adaptive-services service-package extension-provider object-cache-size
object-cache-size
```

In this example, the size of the object cache is 1280 MB.

```
 [edit chassis fpc 1 pic 0]
user@host# set adaptive-services service-package extension-provider object-cache-size 1280
```

65. Configure the size of the policy database in megabytes.

```
 [edit chassis fpc slot pic slot]
user@host# set adaptive-services service-package extension-provider policy-db-size policy-db-size
```

In this example, the size of the policy database is 64 MB.

```
 [edit chassis fpc 1 pic 0]
user@host# set adaptive-services service-package extension-provider policy-db-size 64
```

66. Configure the packages.

```
 [edit chassis fpc slot pic slot]
user@host# set adaptive-services service-package extension-provider package package
```

In this example, the first package is **jservices-appid**, the second package is **jservices-aacl**, the third package is **jservices-llpdf**, the fourth package is **jservices-idp**, and the fifth package is **jservices-sfw**. jservices-sfw is available only in Junos OS Release 10.1 and later.

```
 [edit chassis fpc 1 pic 0]
user@host# set adaptive-services service-package extension-provider package jservices-appid
user@host# set adaptive-services service-package extension-provider package jservices-aacl
user@host# set adaptive-services service-package extension-provider package jservices-llpdf
user@host# set adaptive-services service-package extension-provider package jservices-idp
user@host# set adaptive-services service-package extension-provider package jservices-sfw
```

67. Configure the IP network services.

```
 [edit chassis]
user@host# set network-services ip
```

68. Go to the following hierarchy level and verify the configuration:

```
[edit chassis]
user@host# show chassis
no-service-pic-restart-on-failover;
filter-memory-enhanced;
redundancy {
    graceful-switchover;
}
fpc 0 {
    pic 1 {
        adaptive-services {
            service-package {
                extension-provider {
                    control-cores 1;
                    data-cores 7;
                    object-cache-size 1280;
                    policy-db-size 64;
                    package jservices-appid;
                    package jservices-aacl;
                    package jservices-llpdf;
                    package jservices-idp;
                    package jservices-sfw;
                }
            }
        }
    }
}
network-services ip;
```

Example: Virtual Routing and Forwarding (VRF) and Service Configuration

The following example combines virtual routing and forwarding (*VRF*) and services configuration:

```
[edit policy-options]
policy-statement test-policy {
    term t1 {
        then reject;
    }
}
[edit routing-instances]
```

```
test {
    interface ge-0/2/0.0;
    interface sp-1/3/0.20;
    instance-type vrf;
    route-distinguisher 10.58.255.1:37;
    vrf-import test-policy;
    vrf-export test-policy;
    routing-options {
        static {
            route 0.0.0.0/0 next-table inet.0;
        }
    }
}
[edit interfaces]
ge-0/2/0 {
    unit 0 {
        family inet {
            service {
                input service-set nat-me;
                output service-set nat-me;
            }
        }
    }
}
sp-1/3/0 {
    unit 0 {
        family inet;
    }
    unit 20 {
        family inet;
        service-domain inside;
    }
    unit 21 {
        family inet;
        service-domain outside;
    }
    [edit services]
    stateful-firewall {
        rule allow-any-input {
            match-direction input;
            term t1 {
                then accept;
            }
        }
    }
    nat {
        pool hide-pool {
            address 10.58.16.100;
            port automatic;
        }
        rule hide-all-input {
            match-direction input;
            term t1 {
                then {
                    translated {
                        source-pool hide-pool;
                        translation-type source napt-44;
                    }
                }
            }
        }
    }
    service-set nat-me {
        stateful-firewall-rules allow-any-input;
        nat-rules hide-all-input;
        interface-service {
            service-interface sp-1/3/0.20;
        }
    }
}
```

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
17.1R1	The IDP functionality is deprecated for the MX Series for Junos OS release 17.1R1 and above.
17.1R1	The [edit security idp] statements are deprecated for the MX Series for Junos OS release 17.1R1 and above.
17.1	`accept skip-ids`—The packet is accepted and sent on to its destination, but IDS rule processing configured on an MS-MPC is skipped.

14.2	Starting in Junos OS Release 14.2, MS-MPC and MS-MIC interface cards support IPv6 traffic for Junos Network Secure Stateful Firewall.

Monitoring Stateful Firewalls

IN THIS SECTION

- Monitoring Stateful Firewall Conversations | 586
- Monitoring Global Stateful Firewall Statistics | 587

Monitoring Stateful Firewall Conversations

IN THIS SECTION

- Purpose | 586
- Action | 586

Purpose

Use the `show services stateful-firewall conversations` command to show conversations, or collections of related flows.

Action

```
user@host# show services stateful-firewall conversations
Interface: sp-0/0/0, Service set: sset
Conversation: ALG protocol: tcp
Number of initiators: 1, Number of responders: 1
Flow State Dir Frm
count
TCP 10.0.0.1:1025 -> 128.0.0.1:80 Forward I 372755
NAT source 10.0.0.1:1025 -> 129.0.0.1:1024
```

```
Softwire 2001:0:0:1::1 -> 1001::1
TCP 128.0.0.1:80 -> 129.0.0.1:1024 Forward O 794083
NAT dest 129.0.0.1:1024 -> 10.0.0.1:1025
Softwire 2001:0:0:1::1 -> 1001::1
```

Monitoring Global Stateful Firewall Statistics

IN THIS SECTION

- Purpose | 587
- Action | 587

Purpose

Use the `show services stateful-firewall statistics` command to observe statistics for service sets containing softwire rules.

Action

```
user@host# show services stateful-firewall statistics
Interface Service set Accept Discard Reject Errors
sp-0/0/0 dslite-svc-set2 118991296 0 0 0
sp-0/1/0 dslite-svc-set1 237615050 0 0 0
```

CHAPTER 27

IDS on MS-DPC

IN THIS CHAPTER

- IDS on MS-DPC | 588

IDS on MS-DPC

IN THIS SECTION

- Understanding SYN Cookie Protection on an MS-DPC | 588
- Configuring IDS Rules on an MS-DPC | 590
- Configuring IDS Rule Sets on an MS-DPC | 600
- Examples: Configuring IDS Rules on an MS-DPC | 601

Understanding SYN Cookie Protection on an MS-DPC

SYN cookie is a stateless SYN proxy mechanism you can use in conjunction with other defenses against a SYN flood attack. SYN cookie is supported on the MS-DPC multiservices card.

As with traditional SYN proxying, SYN cookie is activated when the SYN flood attack threshold is exceeded. However, because SYN cookie is stateless, it does not set up a session or policy and route lookups upon receipt of a SYN segment, and it maintains no connection request queues. This dramatically reduces CPU and memory usage and is the primary advantage of using SYN cookie over the traditional SYN proxying mechanism.

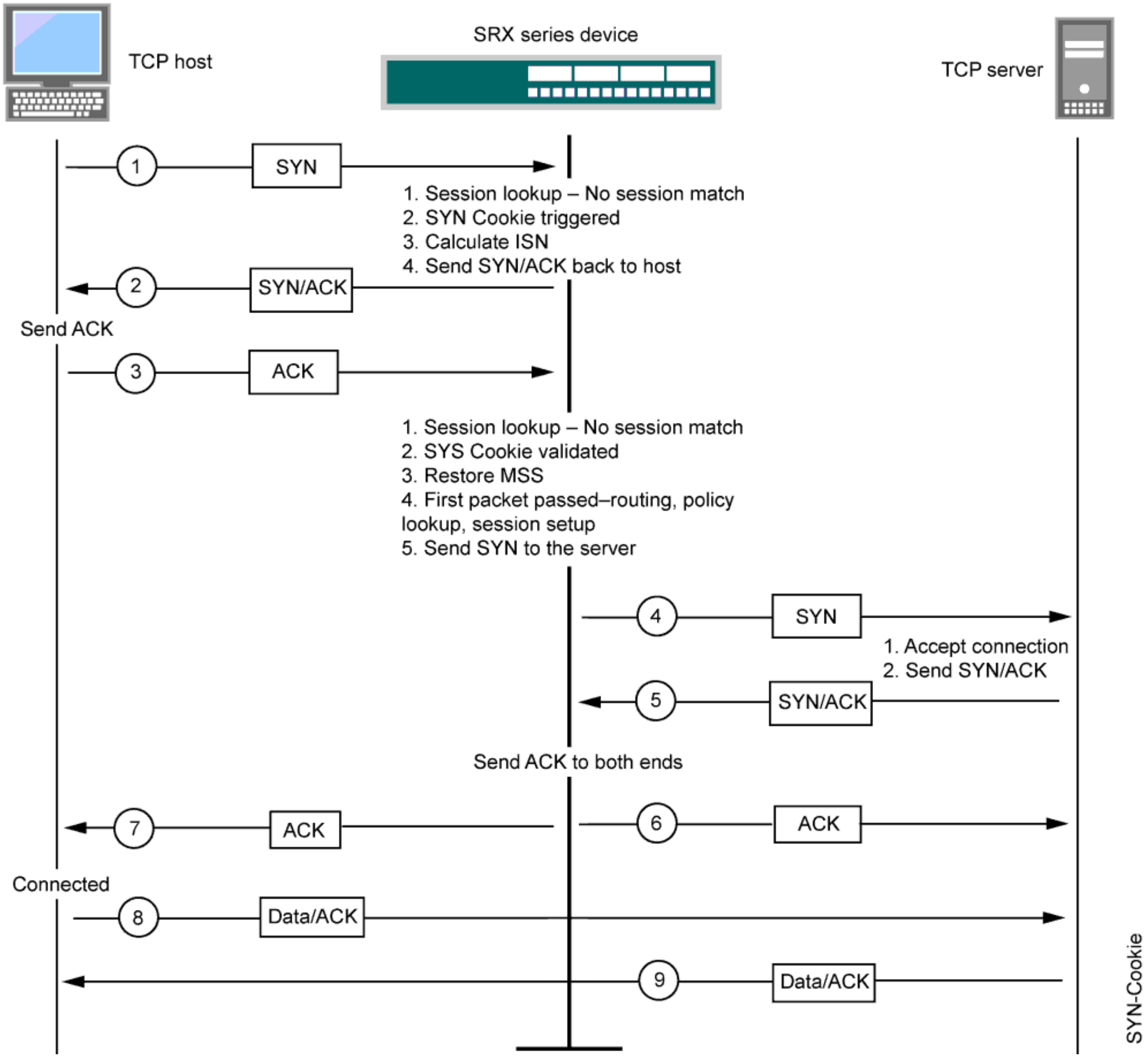
When SYN cookie is enabled on Junos OS and becomes the TCP-negotiating proxy for the destination server, it replies to each incoming SYN segment with a SYN/ACK containing an encrypted cookie as its initial sequence number (ISN). The cookie is an MD5 hash of the original source address and port number, destination address and port number, and ISN from the original SYN packet. After sending the cookie, Junos OS drops the original SYN packet and deletes the calculated cookie from memory. If there is no response to the packet containing the cookie, the attack is noted as an active SYN attack and is effectively stopped.

If the initiating host responds with a TCP packet containing the cookie +1 in the TCP ACK field, Junos OS extracts the cookie, subtracts 1 from the value, and recomputes the cookie to validate that it is a legitimate ACK. If it is legitimate, Junos OS starts the TCP proxy process by setting up a session and sending a SYN to the server containing the source information from the original SYN. When Junos OS receives a SYN/ACK from the server, it sends ACKs to the server and to the initiation host. At this point the connection is established and the host and server are able to communicate directly.

NOTE: The use of SYN cookie or SYN proxy enables the router device to protect the TCP servers behind it from SYN flood attacks in IPv6 flows.

Figure 32 on page 590 shows how a connection is established between an initiating host and a server when SYN cookie is active on Junos OS.

Figure 32: Establishing a Connection with SYN Cookie Active



Configuring IDS Rules on an MS-DPC

IN THIS SECTION

- Configuring Match Direction for IDS Rules | 592
- Configuring Match Conditions in IDS Rules | 593
- Configuring Actions in IDS Rules | 594

IDS rules configured with an MS-DPC identify traffic for which you want the router software to count events. Because IDS is based on stateful firewall properties, you must configure at least one stateful firewall rule and include it in the service set with the IDS rules; for more information, see "Configuring Stateful Firewall Rules" on page 550 .

NOTE: To configure network attack protection with an MS-MPC, see "Configuring Protection Against Network Attacks on an MS-MPC" on page 610 .

To configure an IDS rule, include the `rule` *`rule-name`* statement at the `[edit services ids]` hierarchy level:

```
[edit services ids]
rule rule-name {
    match-direction (input | output | input-output);
    term term-name {
        from {
            application-sets set-name;
            applications [ application-names ];
            destination-address (address | any-unicast) <except>;
            destination-address-range low minimum-value high maximum-value <except>;
            destination-prefix-list list-name <except>;
            source-address (address | any-unicast) <except>;
            source-address-range low minimum-value high maximum-value <except>;
            source-prefix-list list-name <except>;
        }
        then {
            aggregation (IDS) {
                destination-prefix prefix-value | destination-prefix-ipv6 prefix-value;
                source-prefix prefix-value | source-prefix-ipv6 prefix-value;
            }
            (force-entry | ignore-entry);
            logging {
                syslog;
                threshold rate;
            }
            session-limit {
                by-destination (IDS MS-DPC) {
                    hold-time seconds;
                    maximum number;
                    packets number;
                    rate number;
                }
                by-pair (IDS MS-DPC) {
                    hold-time seconds;
                    maximum number;
                    packets number;
                    rate number;
                }
                by-source (IDS MS-DPC) {
                    hold-time seconds;
                    maximum number;
                    packets number;
                    rate number;
                }
            }
            syn-cookie {
                mss value;
                threshold rate;
            }
        }
    }
}
```

Each IDS rule consists of a set of terms, similar to a filter configured at the `[edit firewall]` hierarchy level. A term consists of the following:

- `from` statement—Specifies the match conditions and applications that are included and excluded.
- `then` statement—Specifies the actions and action modifiers to be performed by the router software.

The following sections describe IDS rule content in more detail:

Configuring Match Direction for IDS Rules

Each rule must include a `match-direction` statement that specifies whether the match is applied on the input or output side of the interface. To configure where the match is applied, include the `match-direction (input | input-output | output)` statement at the `[edit services ids rule rule-name]` hierarchy level:

```
[edit services ids rule rule-name]
match-direction (input | output | input-output);
```

If you configure `match-direction input-output`, bidirectional rule creation is .

The match direction is used with respect to the traffic flow through the AS or Multiservices PIC. When a packet is sent to the PIC, direction information is carried along with it.

With an interface service set, packet direction is determined by whether a packet is entering or leaving the interface on which the service set is applied.

With a next-hop service set, packet direction is determined by the interface used to route the packet to the AS or Multiservices PIC. If the inside interface is used to route the packet, the packet direction is input. If the outside interface is used to direct the packet to the PIC, the packet direction is output. For more information on inside and outside interfaces, see "Configuring Service Sets to be Applied to Services Interfaces" on page 10 .

On the AS or Multiservices PIC, a flow lookup is performed. If no flow is found, rule processing is performed. All rules in the service set are considered. During rule processing, the packet direction is compared against rule directions. Only rules with direction information that match the packet direction are considered.

Configuring Match Conditions in IDS Rules

To configure IDS match conditions, include the `from` statement at the `[edit services ids rule` *`rule-name`* `term` *`term-name`*`]` hierarchy level:

```
[edit services ids rule rule-name term term-name]
from {
    application-sets set-name;
    applications [ application-names ];
    destination-address (address | any-unicast) <except>;
    destination-address-range low minimum-value high maximum-value <except>;
    destination-prefix-list list-name <except>;
    source-address (address | any-unicast) <except>;
    source-address-range low minimum-value high maximum-value <except>;
    source-prefix-list list-name <except>;
}
```

If you omit the `from` statement, the software accepts all events and places them in the IDS cache for processing.

The source address and destination address can be either IPv4 or IPv6. You can use the destination address, a range of destination addresses, a source address, or a range of source addresses as a match condition, in the same way that you would configure a firewall filter; for more information, see the Routing Policies, Firewall Filters, and Traffic Policers User Guide.

Alternatively, you can specify a list of source or destination prefixes by including the `prefix-list` statement at the `[edit policy-options]` hierarchy level and then including either the `destination-prefix-list` or `source-prefix-list` statement in the IDS rule. For an example, see "Examples: Configuring Stateful Firewall Rules" on page 557 .

You can also include application protocol definitions that you have configured at the `[edit applications]` hierarchy level; for more information, see "Configuring Application Properties" on page 508 .

- To apply one or more specific application protocol definitions, include the `applications` statement at the `[edit services ids rule` *`rule-name`* `term` *`term-name`* `from]` hierarchy level.
- To apply one or more sets of application protocol definitions that you have defined, include the `application-sets` statement at the `[edit services ids rule` *`rule-name`* `term` *`term-name`* `from]` hierarchy level.

> **NOTE**: If you include one of the statements that specifies application protocols, the router derives port and protocol information from the corresponding configuration at the `[edit applications]` hierarchy level; you cannot specify these properties as match conditions.

If a match occurs on an application, the application protocol is displayed separately in the `show services ids` command output. For more information, see the CLI Explorer.

Configuring Actions in IDS Rules

To configure IDS actions, include the `then` statement at the `[edit services ids rule` *`rule-name`* `term` *`term-name`*`]` hierarchy level:

```
[edit services ids rule rule-name term term-name]
then {
    aggregation (IDS) {
        destination-prefix prefix-value | destination-prefix-ipv6 prefix-value;
        source-prefix prefix-value | source-prefix-ipv6 prefix-value;
    }
    (force-entry | ignore-entry);
    logging {
        syslog;
        threshold rate;
    }
    session-limit {
        by-destination (IDS MS-DPC) {
            hold-time seconds;
            maximum number;
            packets number;
            rate number;
        }
        by-pair (IDS MS-DPC) {
            hold-time seconds;
            maximum number;
            packets number;
            rate number;
        }
        by-source (IDS MS-DPC) {
            hold-time seconds;
            maximum number;
            packets number;
            rate number;
        }
    }
    syn-cookie {
        mss value;
        threshold rate;
    }
}
```

You can configure the following possible actions:

- `aggregation`—The router aggregates traffic labeled with the specified source or destination prefixes before passing the events to IDS processing. This is helpful if you want to examine all the traffic connected with a particular source or destination host. To collect traffic with some other marker, such as a particular application or port, configure that value in the match conditions.

  To configure aggregation prefixes, include the `aggregation` statement at the `[edit services ids rule` *`rule-name`* `term` *`term-name`* `then]` hierarchy level and specify values for `source-prefix`, `destination-prefix` `source-prefix-ipv6`, or `destination-prefix-ipv6`:

  ```
  [edit services ids rule rule-name term term-name then]
      aggregation (IDS) {
      destination-prefix prefix-value | destination-prefix-ipv6 prefix-value;
      source-prefix prefix-value | source-prefix-ipv6 prefix-value;
  }
  ```

  The value of `source-prefix` and `destination-prefix` must be an integer between 1 and 32. The value of `source-prefix-ipv6` and `destination-prefix-ipv6` must be an integer between 1 and 128.

- `(force-entry | ignore-entry)`—`force-entry` provides a permanent spot in IDS caches for subsequent events after one event is registered. By default, the IDS software does not record information about “good” packets that do not exhibit suspicious behavior. You can use the `force-entry` statement to record all traffic from a suspect host, even traffic that would not otherwise be counted.

ignore-entry ensures that all IDS events are ignored. You can use this statement to disregard all traffic from a host you trust, including any temporary anomalies that IDS would otherwise count as events.

To configure an entry behavior different from the default, include the force-entry or ignore-entry statement at the [edit services ids rule *rule-name* term *term-name* then] hierarchy level:

```
[edit services ids rule rule-name term term-name then]
(force-entry | ignore-entry);
```

- logging—The event is logged in the system log file.

  To configure logging, include the logging statement at the [edit services ids rule *rule-name* term *term-name* then] hierarchy level:

```
[edit services ids rule rule-name term term-name then]
logging {
    syslog;
    threshold rate;
}
```

  You can optionally include a threshold rate to trigger the generation of system log messages. The threshold rate is specified in events per second. IDS logs are generated once every 60 seconds for each anomaly that is reported. The logs are generated as long as the events continue.

- session-limit—The router limits open sessions when the specified threshold is reached.

  To configure a threshold, include the session-limit statement at the [edit services ids rule *rule-name* term *term-name* then] hierarchy level:

```
[edit services ids rule rule-name term term-name then]
session-limit {
    by-destination (IDS MS-DPC) {
        hold-time seconds;
        maximum number;
        packets number;
        rate number;
    }
    by-pair (IDS MS-DPC) {
        hold-time seconds;
        maximum number;
        packets number;
        rate number;
    }
    by-source (IDS MS-DPC) {
        hold-time seconds;
        maximum number;
        packets number;
        rate number;
    }
}
```

You configure the thresholds for flow limitation based on traffic direction:

- To limit the number of outgoing sessions from one internal host or subnet, configure the `by-source` statement.
- To limit the number of sessions between a pair of IP addresses, subnets, or applications, configure the `by-pair` statement.
- To limit the number of incoming sessions to one external public IP address or subnet, configure the `by-destination` statement.

For each direction, you can configure the following threshold values:

- `hold-time` *`seconds`*—When the `rate` or `packets` measurement reaches the threshold value, stop all new flows for the specified number of seconds. Once `hold-time` is in effect, the traffic is blocked for the specified time even if the rate subsides below the specified limit. By default, `hold-time` has a value of 0; the range is 0 through 60 seconds.
- `maximum` *`number`*—Maximum number of open sessions per IP address or subnet per application. The range is 1 through 32,767.
- `packets` *`number`*—Maximum number of packets per second (pps) per IP address or subnet per application. The range is 4 through 2,147,483,647.
- `rate` *`number`*—Maximum number of sessions per second per IP address or subnet per application. The range is 4 through 32,767.

  If you include more than one source address in the match conditions configured at the `[edit services ids rule` *`rule-name`* `term` *`term-name`* `from]` hierarchy level, limits are applied for each source address independently. For example, the following configuration allows 20 connections from each source address (`10.1.1.1` and `10.1.1.2`), not 20 connections total. The same logic applies to the `applications` and `destination-address` match conditions.

  ```
  [edit services ids rule rule-name term term-name]
      from {
      source-address 10.1.1.1;
  ```

```
        source-address 10.1.1.2;
        }
        then {
        session-limit by-source {
            maximum 20;
        }
    }
```

NOTE: IDS limits are applied to packets that are accepted by stateful firewall rules. They are not applied to packets discarded or rejected by stateful firewall rules. For example, if the stateful firewall accepts 75 percent of the incoming traffic and the remaining 25 percent is rejected or discarded, the IDS limit applies only to 75 percent of the traffic.

- `syn-cookie`—The router activates SYN-cookie defensive mechanisms.

  To configure SYN-cookie values, include the `syn-cookie` statement at the `[edit services ids rule` *`rule-name`* `term` *`term-name`* `then]` hierarchy level:

  ```
  [edit services ids rule rule-name term term-name then]
  syn-cookie {
      mss value;
      threshold rate;
  }
  ```

  If you enable SYN-cookie defenses, you must include both a threshold rate to trigger SYN-cookie activity and a Transmission Control Protocol (TCP) maximum segment size (MSS) value for TCP delayed binding. The threshold rate is specified in SYN attacks per second. By default, the TCP MSS value is 1500; the range is from 128 through 8192.

  Handling of SYN Flood Attacks and SYN Cookie Protection

  The main purpose of a SYN flood attack is to consume all new network connections at a site and thereby prevent authorized and legitimate users from being able to connect to network resources. The SYN (synchronize sequence number) packet is the first request to connect sent to a system. The SYN packet contains an ID to which the receiver is required to respond. If the packet contains an illegal ID, the receiving system does receive a connection acknowledgment when it responds to the intended connection initiator. Eventually, this half-open connection times out and the incoming channel on the receiver becomes available again to normally handle another request. A SYN flood attack sends so many such requests that all incoming connections are continuously tied up waiting for acknowledgments that are never received. This condition causes the server to be unavailable to legal users (except in cases where a user session is established when it is exactly at the moment when one of the tied-up connections times out). A SYN flood attack is a connectionless attack. It does not require a real source IP addresses and, because it uses legitimate destination IP or port addresses, is practically impossible to distinguish from legitimate packets. Therefore, it is very difficult to prevent this type of attack by using only filters or stateful firewall rules. Basically, there are only three methods to protect from this type of attack:

- Intercept (delayed binding)—The firewall intercepts incoming TCP synchronization requests and establishes a connection with the client on the server's behalf, and with the server on the client's behalf. If both connections are successful, the firewall transparently merges the two connections. The firewall usually has aggressive timeouts to prevent its own resources from being consumed by a SYN attack. This the most intensive solution in terms of processing and memory requirements.

- Watch (SYN defense)—The firewall passively watches half-open connections and actively closes connections on the server after a configurable length of time.

- SYN cookie—SYN cookies are particular choices for the initial TCP sequence number chosen by the TCP server. A host requesting a connection must answer with the cookie to connect to an open TCP socket while a SYN-flood has been detected as in progress by the IDS.

Juniper Networks routers support the combination of stateful firewall and IDS mechanisms to support the SYN cookie and watch (SYN defense) methods. The key to the SYN flood attack is the filling of the SYN queue of the victim or the attacked network element. The SYN cookie defense method enables the victim to continue accepting connection requests when the SYN queue is full or, in the case of the firewall or IDS applications, when a certain threshold has been reached. After the threshold is reached, a cryptographic cookie (a 32-bit number) is created from information in the SYN segment and the SYN segment is dropped. The cookie is used as the initial sequence number in the SYN-ACK sent to the client. The cookie (plus one) is returned to the firewall or IDS application as the acknowledgment number in the ACK from a legitimate client. The returned cookie can be validated and the most important parts of the SYN segment can be reconstructed from the cookie, thereby allowing a connection to be established. Because the spoofed clients of the SYN flood never send ACKs, no resources are allocated for them in any state when SYN cookies are in use. It is preferred that you use SYN flood countermeasures only for hosts under attack. The anomaly table can be used for reliable attack recognition or they can be enabled within the stateful firewall. Such a type of configuration also helps prevent the depletion of system resources (especially the flow table) in case of attacks.

When combining multiple services, the general path is an important factor for consideration in the forward and reverse directions. This is especially true when NAT is deployed to determine whether the pre-NAT or post-NAT address must be used to match a rule. In the forward path from a LAN interface to a WAN interface, IDS and stateful firewall are performed first, then NAT, and finally IPSec. This sequence of processing of services denotes that the stateful firewall must match on a pre-NAT address, whereas the IPSec tunnel matches on the post-NAT address. In the return path, the IPSec packet is processed first, then NAT, and finally the stateful firewall. This order of processing still allows IPSec to match a public address and the stateful firewall to match on a private address. You must separately configure the firewall, NAT, and IDS services. The processing of packets becomes much more complicated when IPSec over GRE is implemented in the router with other services turned on. This behavior occurs because Junos OS treats GRE packets in a unique fashion after GRE encapsulation. After a packet is encapsulated in a GRE packet, it is marked with an input interface as the next-hop outgoing interface. This method of marking causes GRE packets to be blocked if any input filters or input services are that do not allow for this service.

Junos OS services support a limited set of IDS rules to help detect attacks such as port scanning and anomalies in traffic patterns. It also supports some attack prevention by limiting the number of flows, sessions, and rates. In addition, it protects against SYN attacks by implementing a SYN cookie mechanism. Because the intrusion detection and prevention (IDP) service does not support higher-layer application signatures, an effective approach against attacks is that protection against a SYN attack can be configured. The IDP solution is largely a monitoring tool and not an essential prevention tool. To prevent a SYN attack, the router will operate as a type of SYN “proxy” and utilizes cookie values. When this feature is turned on, the router responds to the initial SYN packet with a SYN-ACK packet that contains a unique cookie value in the sequence number field. If the initiator responds with the same cookie in the sequence field, the TCP flow is accepted; if the responder does not respond or if it responds with the wrong cookie, the flow is dropped. To trigger this defense, you must configure a SYN cookie threshold. To enable the SYN cookie defense, an IDS rule action must contain a threshold that indicates when the feature should be enabled and an MSS value to avoid having the router manage segmented fragments when acting as a SYN proxy:

```
[edit]
user@host# set services ids rule simple-ids term 1 then syn-cookie
```

Configuring IDS Rule Sets on an MS-DPC

You can use `rule-set` statement to define a collection of IDS rules that determine what actions the router software performs on packets in the data stream. This is supported on the MS-DPC multiservices card. (To configure network attack protection with an MS-MPC, see "Configuring Protection Against Network Attacks on an MS-MPC" on page 610 .)

You define each rule by specifying a rule name and configuring terms. Then, you specify the order of the rules by including the `rule-set` statement at the `[edit services ids]` hierarchy level with a `rule` statement for each rule:

```
[edit services ids]
rule-set rule-set-name {
    rule rule-name;
}
```

The router software processes the rules in the order in which you specify them in the configuration. If a term in a rule matches the packet, the router performs the corresponding action and the rule processing stops. If no term in a rule matches the packet, processing continues to the next rule in the rule set. If none of the rules matches the packet, the packet is dropped by default.

Examples: Configuring IDS Rules on an MS-DPC

The following configuration adds a permanent entry to the IDS anomaly table when it encounters a flow with the destination address 10.410.6.2. This example is supported on the MS-DPC multiservices card. (To configure network attack protection with an MS-MPC, see "Configuring Protection Against Network Attacks on an MS-MPC" on page 610 .)

```
[edit services ids]
rule simple_ids {
    term 1 {
        from {
            destination-address 10.410.6.2/32;
        }
        then {
            force-entry;
            logging {
                threshold 1;
                syslog;
            }
        }
    }
    term default {
        then {
            aggregation {
                source-prefix 24;
            }
        }
    }
    match-direction input;
}
```

The IDS configuration works in conjunction with the stateful firewall mechanism and relies heavily on the anomalies reported by the stateful firewall. The following configuration example shows this relationship:

```
[edit services ids]
rule simple_ids {
    term 1 {
        from {
            source-address 10.30.20.2/32;
            destination-address {
                10.30.10.2/32;
                10.30.1.2/32 except;
            }
            applications appl-ftp;
        }
        then {
            force-entry;
            logging {
                threshold 5;
                syslog;
            }
            syn-cookie {
                threshold 10;
            }
        }
    }
    match-direction input;
}
```

```
[edit services stateful-firewall]
rule my-firewall-rule {
    match-direction input-output;
    term term1 {
        from {
            source-address  10.30.20.2/32;
            applications appl-ftp ;
            destination-address {
                10.30.10.2/32;
                10.30.1.2/32 except;
            }
        }
        then {
            accept;
            syslog;
        }
    }
}
```

The stateful firewall or NAT service is used to generate the input data for the IDS application. When you enable and configure an IDS service, you must also enable stateful firewall with at least one rule (accept or discard all traffic). When the system is under an attack, the stateful firewall sends the correct and complete list of attack events to the IDS system. In your network environment, you can ensure that the system is wholly protected against a whole range of attacks so that the IDS system reports all such attacks. You must exercise caution when you configure the system to be protected from all attacks and unauthenticated access scenarios so that the traffic bandwidth that the system handles is not burdened. It is also important to verify the correlation between the firewall syslog messages corresponding to the attacks and IDS tables. The IDS tables must have the same or slightly less number of anomalies or errors compared to the firewall-based syslog messages. You can use the appropriate show commands are used to display the IDS tables.

A default stateful firewall rule can be as simple as only allowing connection initiation from the inside interface to the outside interface and discarding all other packets. However, in a real-word network environment, rules are generally more complex, such as configuring only a certain tributary unit ports are to be opened, using application layer gateways (ALGs) for complicated protocols, and using NAT for both outgoing connections and inside hosts such as HTTP servers. Therefore, it is necessary to also configure the system as needed to interwork with simple and complicated rules. For example, if a SYN attack is directed towards an inside address that is simply discarded, no anomalies need to be reported to the IDS system. But if the SYN attack is directed towards the real HTTP server, anomalies must be reported. The IDS system can mitigate SYN attacks by using the TCP SYN cookie defense capability. You can enable the SYN cookie protection methodology by setting a threshold for SYNs per second for a given host and also a maximum segment size (MSS). Because the IDS system uses the stateful firewall, a firewall rule must be defined in the service-set. If you do not configure the `from` statement in a stateful firewall (rule term match condition) at the `[edit services service-set service-set-name stateful-firewall-rules` *`rule-name`* `term` *`term-name`*`]` hierarchy level, it signifies that all events are placed into the IDS cache.

The following example shows configuration of flow limits:

```
[edit services ids]
rule ids-all {
    match-direction input;
    term t1 {
        from {
            application-sets alg-set;
        }
        then {
            aggregation {
                destination-prefix 30; /* IDS action aggregation */
            }
            logging {
                threshold 10;
            }
            session-limit {
                by-destination {
                    hold-time 0;
                    maximum 10;
                    packets 200;
                    rate 100;
                }
                by-pair {
                    hold-time 0;
                    maximum 10;
                    packets 200;
                    rate 100;
                }
                by-source {
                    hold-time 5;
                    maximum 10;
                    packets 200;
                    rate 100;
                }
            }
        }
    }
}
```

CHAPTER 28

Network Attack Protection on MS-MPC and MS-MIC

IN THIS CHAPTER

- Network Attack Protection on MS-MPC and MS-MIC | 605

Network Attack Protection on MS-MPC and MS-MIC

IN THIS SECTION

- Understanding IDS on an MS-MPC | 605
- Configuring Protection Against Network Attacks on an MS-MPC | 610
- Configuring Logging of Network Attack Protection Packet Drops on an MS-MPC | 621

Understanding IDS on an MS-MPC

IN THIS SECTION

- Intrusion Detection Services | 606
- Benefits | 606
- Session Limits | 606
- Suspicious Packet Patterns | 607
- Header Anomaly Attacks | 609

Intrusion Detection Services

Intrusion detection services (IDS) rules on an MS-MPC give you a way to identify and drop traffic that is part of a network attack.

IDS rules provide a more granular level of filtering than firewall filters and policers, which can stop illegal TCP flags and other bad flag combinations, and can enforce general rate limiting (see the *Routing Policies, Firewall Filters, and Traffic Policers User Guide*). You can use firewall filters and policers along with IDS to reduce the traffic that needs to be processed by an IDS rule.

In an IDS rule, you can specify:

- Limits on the sessions that originate from individual sources or that terminate at individual destinations. This protects against network probing and flooding attacks.
- Types of suspicious packets to drop.

To protect against header anomaly attacks, a header integrity check is automatically performed if you configure an IDS rule, stateful firewall rule, or a NAT rule and apply it to the service set. You can also explicitly configure a header integrity check for the service set if you do not assign the service set an IDS rule, stateful firewall rule, or a NAT rule.

Benefits

- Provides protection against several types of network attacks.

Session Limits

You can use IDS rules to set session limits for traffic from an individual source or to an individual destination. This protects against network probing and flooding attacks. Traffic that exceeds the session limits is dropped. You can specify session limits either for traffic with a particular IP protocol, such as ICMP, or for traffic in general.

You decide whether the limits apply to individual addresses or to an aggregation of traffic from individual subnets of a particular prefix length. For example, if you aggregate limits for IPv4 subnets with a prefix length of 24, traffic from 192.0.2.2 and 192.0.2.3 is counted against the limits for the 192.0.2.0/24 subnet.

Some common network probing and flooding attacks that session limits protect against include:

ICMP Address Sweep

The attacker sends ICMP request probes (pings) to multiple targets. If a target machine replies, the attacker receives the IP address of the target.

ICMP Flood The attacker floods a target machine by sending a large number of ICMP packets from one or more source IP addresses. The target machine uses up its resources as it attempts to process those ICMP packets, and can no longer process valid traffic.

TCP Port Scan The attacker sends TCP SYN packets from one source to multiple destination ports of the target machine. If the target replies with a SYN-ACK from one or more destination ports, the attacker learns which ports are open on the target.

TCP SYN Flood The attacker floods a target machine by sending a large number of TCP SYN packets from one or more source IP addresses. The attacker might use real source IP addresses, which results in a completed TCP connection, or might use fake source IP addresses, resulting in the TCP connection not being completed. The target creates states for all the completed and uncompleted TCP connections. The target uses up its resources as it attempts to manage the connection states, and can no longer process valid traffic.

UDP Flood The attacker floods a target machine by sending a large number of UDP packets from one or more source IP addresses. The target machine uses up its resources as it attempts to process those UDP packets, and can no longer process valid traffic.

Session limits for traffic from a source or to a destination include:

- maximum number of concurrent sessions
- maximum number of packets per second
- maximum number of connections per second

IDS also installs a dynamic filter on the PFEs of line cards for suspicious activity when the following conditions occur:

- Either the packets per second or the number of connections per second for an individual source or destination address (not for a subnet) exceeds four times the session limit in the IDS rule. This session limit is the general source or destination limit for the IDS rule, not the limit specified for a particular protocol.
- The services card CPU utilization percentage exceeds a configured value (default value is 90 percent).

The dynamic filter drops the suspicious traffic at the PFE, and the traffic is not sent to the MS-MPC to be processed by the IDS rule. When the packet or connection rate no longer exceeds four times the limit in the IDS rule, the dynamic filter is removed.

Suspicious Packet Patterns

You can use IDS rules to identify and drop traffic with a suspicious packet pattern. This protects against attackers that craft unusual packets to launch denial-of-service attacks.

Suspicious packet patterns and attacks that you can specify in an IDS rule are:

ICMP fragmentation attack

The attacker sends the target ICMP packets that are IP fragments. These are considered suspicious packets because ICMP packets are usually short. When the target receives these packets, the results can range from processing packets incorrectly to crashing the entire system.

ICMP large packet attack

The attacker sends the target ICMP frames with an IP length greater than 1024 bytes. These are considered suspicious packets because most ICMP messages are small.

ICMP Ping of death attack

The attacker sends the target ICMP ping packets whose IP datagram length (ip_len) exceeds the maximum legal length (65,535 bytes) for IP packets, and the packet is fragmented. When the target attempts to reassemble the IP packets, a buffer overflow might occur, resulting in a system crashing, freezing, and restarting.

IP Bad option attack

The attacker sends the target packets with incorrectly formatted IPv4 options or IPv6 extension headers. This can cause unpredictable issues, depending on the IP stack implementation of routers and the target.

IPv4 options

Attackers can maliciously use IPv4 options for denial-of-service attacks.

IPv6 extension headers

Attackers can maliciously use extension headers for denial-of-service attacks or to bypass filters.

IP teardrop attack

The attacker sends the target fragmented IP packets that overlap. The target machine uses up its resources as it attempts to reassemble the packets, and can no longer process valid traffic.

IP unknown protocol attack

The attacker sends the target packets with protocol numbers greater than 137 for IPv4 and 139 for IPv6. An unknown protocol might be malicious.

Land attack

The attacker sends the target spoofed SYN packets that contain the target's IP address as both the destination and the source IP address. The target uses up its resources as it repeatedly replies to itself. In another variation of the land attack, the SYN packets also contain the same source and destination ports.

SYN fragment attack

The attacker sends the target SYN packet fragments. The target caches SYN fragments, waiting for the remaining fragments to arrive so it can reassemble them and complete the connection. A flood of SYN fragments eventually fills the host's memory buffer, preventing valid traffic connections.

TCP FIN No ACK attack The attacker sends the target TCP packets that have the FIN bit set but have the ACK bit unset. This can allow the attacker to identify the operating system of the target or to identify open ports on the target.

TCP no flag attack The attacker sends the target TCP packets containing no flags. This can cause unpredictable behavior on the target, depending on its TCP stack implementation.

TCP SYN FIN attack The attacker sends the target TCP packets that have both the SYN and the FIN bits set. This can cause unpredictable behavior on the target, depending on its TCP stack implementation.

TCP WinNuke attack The attacker sends a TCP segment with the urgent (URG) flag set and destined for port 139 of a target running Windows. This might cause the target machine to crash.

Header Anomaly Attacks

To protect against header anomaly attacks, a header integrity check is automatically performed if you configure an IDS rule, a stateful firewall rule, or a NAT rule and apply it to the service set. You can also explicitly configure a header integrity check for the service set if you do not assign the service set an IDS rule, stateful firewall rule, or a NAT rule.

The header integrity check provides protection against the following header anomaly attacks:

ICMP Ping of death attack The attacker sends the target ICMP ping packets whose IP datagram length (ip_len) exceeds the maximum legal length (65,535 bytes) for IP packets, and the packet is fragmented. When the target attempts to reassemble the IP packets, a buffer overflow might occur, resulting in a system crashing, freezing, and restarting.

IP unknown protocol attack The attacker sends the target packets with protocol numbers greater than 137 for IPv4 and 139 for IPv6. An unknown protocol might be malicious.

TCP no flag attack The attacker sends the target TCP packets containing no flags. This can cause unpredictable behavior on the target, depending on its TCP stack implementation.

TCP SYN FIN attack The attacker sends the target TCP packets that have both the SYN and the FIN bits set. This can cause unpredictable behavior on the target, depending on its TCP stack implementation.

TCP FIN No ACK attack The attacker sends the target TCP packets that have the FIN bit set but have the ACK bit unset. This can allow the attacker to identify the operating system of the target or to identify open ports on the target.

Configuring Protection Against Network Attacks on an MS-MPC

IN THIS SECTION

- Configuring Protection Against Network Probing, Network Flooding, and Suspicious Pattern Attacks | 610
- Configuring Protection Against Header Anomaly Attacks | 620

This topic includes the following tasks, which describe how to protect against network attacks when using an MS-MPC:

Configuring Protection Against Network Probing, Network Flooding, and Suspicious Pattern Attacks

IN THIS SECTION

- Configuring IDS Rule Name and Direction | 611
- Configuring Session Limits for Subnets | 611
- Configuring Session Limits Independent of the Protocol | 613
- Configuring ICMP Address Sweep Protection | 614
- Configuring TCP Port Scanner Protection | 614
- Configuring ICMP Flooding Protection | 615
- Configuring UDP Flooding Protection | 616
- Configuring TCP SYN Flooding Protection | 616
- Configuring ICMP Fragmentation Protection | 618
- Configuring ICMP Large Packet Protection | 618
- Configuring IP Bad Options Protection | 618
- Configuring Land Attack Protection | 619
- Configuring TCP SYN Fragment Protection | 619
- Configuring WinNuke Protection | 619
- Configuring the Service Set | 620

You configure protection against network probing attacks, network flooding attacks, and suspicious pattern attacks by configuring an intrusion detection service (IDS) rule, and then applying that rule to a service set that is on an MS-MPC. Only the first term of an IDS rule is used, and only the first IDS input rule and the first IDS output rule for a service set are used.

Configuring protection against network probing, network flooding, and suspicious pattern attacks includes:

Configuring IDS Rule Name and Direction

For each IDS rule, you must configure a name and the direction of traffic to which it is applied.

To configure the IDS rule name and direction:

1. Specify a name for the IDS rule.

```
[edit services ids]
user@host# set rule rule-name
```

2. Specify whether the IDS rule is applied to input traffic, output traffic, or both.

```
[edit services ids rule rule-name]
user@host# set match-direction (input | input-output |output)
```

Configuring Session Limits for Subnets

If you want to apply session limits to an aggregation of all attacks to or from individual destination or source subnets rather than for individual addresses, configure aggregation.

To configure subnet aggregation:

- If you want to apply session limits to an aggregation of all attacks from within an individual IPv4 subnet, specify the subnet prefix length. The range is from 1 through 32.

  ```
  [edit services ids rule rule-name term term-name then]
  user@host# set aggregation source-prefix prefix-value
  ```

  For example, the following statement configures an IPv4 prefix length of 24, and attacks from 10.1.1.2 and 10.1.1.3 are counted as attacks from the 10.1.1/24 subnet.

  ```
  [edit services ids rule rule1 term term1 then]
  user@host# set aggregation source-prefix 24
  ```

However, if a single host on a subnet generates a large number of network probing or flooding attacks, the flows for the entire subnet might be stopped.

- If you want to apply session limits to an aggregation of all attacks from within an individual IPv6 subnet, specify the subnet prefix length. The range is from 1 through 128.

```
[edit services ids rule rule-name term term-name then]
user@host# set aggregation source-prefix-ipv6 prefix-value
```

For example, the following statement configures an IPv6 prefix length of 64, and attacks from 2001:db8:1234:72a2::2 and 2001:db8:1234:72a2::3 are counted as attacks from the 2001:db8:1234:72a2::/64 subnet.

```
[edit services ids rule rule1 term term1 then]
user@host# set aggregation source-prefix-ipv6 64
```

However, if a single host on a subnet generates a large number of network probing or flooding attacks, the flows for the entire subnet might be stopped.

- If you want to apply session limits to an aggregation of all attacks to an individual IPv4 subnet, specify the subnet prefix length. The range is from 1 through 32.

```
[edit services ids rule rule-name term term-name then]
user@host# set aggregation destination-prefix prefix-value
```

For example, the following statement configures an IPv4 prefix length of 24, and attacks to 10.1.1.2 and 10.1.1.3 are counted as attacks to the 10.1.1/24 subnet.

```
[edit services ids rule rule1 term term1 then]
user@host# set aggregation destination-prefix 24
```

- If you want to apply session limits to an aggregation of all attacks to an individual IPv6 subnet, specify the subnet prefix length. The range is from 1 through 128.

```
[edit services ids rule rule-name term term-name then]
user@host# set aggregation destination-prefix-ipv6 prefix-value
```

For example, the following statement configures an IPv6 prefix length of 64, and attacks to 2001:db8:1234:72a2::2 and 2001:db8:1234:72a2::3 are counted as attacks to the 2001:db8:1234:72a2::/64 subnet.

```
[edit services ids rule rule1 term term1 then]
user@host# set aggregation destination-prefix-ipv6 64
```

Configuring Session Limits Independent of the Protocol

If you want to configure session limits for traffic to an individual destination or from an individual source independent of the protocol, then perform one or more of the following tasks:

- To configure session limits for source IP addresses or subnets independent of a protocol:
  - Configure the maximum number of concurrent sessions allowed from an individual source IP address or subnet.

    ```
    [edit services ids rule rule-name term term-name then]
    user@host# set session-limit by-source maximum number
    ```

  - Configure the maximum number of packets per second allowed from an individual source IP address or subnet.

    ```
    [edit services ids rule rule-name term term-name then]
    user@host# set session-limit by-source packets number
    ```

  - Configure the maximum number of connections per second allowed from an individual source IP address or subnet.

    ```
    [edit services ids rule rule-name term term-name then]
    user@host# set session-limit by-source rate number
    ```

- To configure session limits for destination IP addresses or subnets independent of a protocol:
  - Configure the maximum number of concurrent sessions allowed for an individual destination IP address or subnet.

    ```
    [edit services ids rule rule-name term term-name then]
    user@host# set session-limit by-destination maximum number
    ```

- Configure the maximum number of packets per second allowed for an individual destination IP address or subnet.

  ```
  [edit services ids rule rule-name term term-name then]
  user@host# set session-limit by-destination packets number
  ```

- Configure the maximum number of connections per second allowed for an individual destination IP address or subnet.

  ```
  [edit services ids rule rule-name term term-name then]
  user@host# set session-limit by-destination rate number
  ```

Configuring ICMP Address Sweep Protection

To configure protection against ICMP address sweeps, configure any combination of the maximum allowed ICMP concurrent sessions, packets per second, and connections per second for a source:

- Configure the maximum number of concurrent ICMP sessions allowed from an individual source IP address or subnet.

  ```
  [edit services ids rule rule-name term term-name then]
  user@host# set session-limit by-source by-protocol icmp maximum number
  ```

- Configure the maximum number of ICMP packets per second allowed from an individual source IP address or subnet.

  ```
  [edit services ids rule rule-name term term-name then]
  user@host# set session-limit by-source by-protocol icmp packets number
  ```

- Configure the maximum number of ICMP connections per second allowed from an individual source IP address or subnet.

  ```
  [edit services ids rule rule-name term term-name then]
  user@host# set session-limit by-source by-protocol icmp rate number
  ```

Configuring TCP Port Scanner Protection

To configure protection against TCP port scanner attacks, configure any combination of the maximum allowed TCP concurrent sessions and connections per second for a source or destination:

- Configure the maximum number of concurrent TCP sessions allowed from an individual source IP address or subnet.

```
[edit services ids rule rule-name term term-name then]
user@host# set session-limit by-source by-protocol tcp maximum number
```

- Configure the maximum number of TCP connections per second allowed for an individual source IP address or subnet.

```
[edit services ids rule rule-name term term-name then]
user@host# set session-limit by-source by-protocol tcp rate number
```

- Configure the maximum number of TCP sessions allowed for an individual destination IP address or subnet.

```
[edit services ids rule rule-name term term-name then]
user@host# set session-limit by-destination by-protocol tcp maximum number
```

- Configure the maximum number of TCP connections per second allowed for an individual destination IP address or subnet.

```
[edit services ids rule rule-name term term-name then]
user@host# set session-limit by-destination by-protocol tcp rate number
```

Configuring ICMP Flooding Protection

To configure protection against ICMP flooding attacks, configure any combination of the maximum allowed ICMP concurrent sessions, packets per second, and number of connections per second for a destination:

- Configure the maximum number of concurrent ICMP sessions allowed for an individual destination IP address or subnet.

```
[edit services ids rule rule-name term term-name then]
user@host# set session-limit by-destination by-protocol icmp maximum number
```

- Configure the maximum number of ICMP packets per second allowed for an individual destination IP address or subnet.

```
[edit services ids rule rule-name term term-name then]
user@host# set session-limit by-destination by-protocol icmp packets number
```

- Configure the maximum number of ICMP connections per second allowed for an individual destination IP address or subnet for ICMP.

```
[edit services ids rule rule-name term term-name then]
user@host# set session-limit by-destination by-protocol icmp rate number
```

Configuring UDP Flooding Protection

To configure protection against UDP flooding attacks, configure any combination of the maximum allowed UDP concurrent sessions, packets per second, and connections per second for a destination:

- Configure the maximum number of concurrent UDP sessions allowed for an individual destination IP address or subnet.

```
[edit services ids rule rule-name term term-name then]
user@host# set session-limit by-destination by-protocol udp maximum number
```

- Configure the maximum number of UDP packets per second allowed for an individual destination IPaddress or subnet.

```
[edit services ids rule rule-name term term-name then]
user@host# set session-limit by-destination by-protocol udp packets number
```

- Configure the maximum number of UDP connections per second allowed for an individual destination IP address or subnet.

```
[edit services ids rule rule-name term term-name then]
user@host# set session-limit by-destination by-protocol udp rate number
```

Configuring TCP SYN Flooding Protection

To configure protection against TCP SYN flooding attacks, configure any combination of the maximum allowed TCP concurrent sessions, packets per second, and connections per second for a source or destination. You can also configure the closing of unestablished TCP connections after a timeout:

- Configure the maximum number of concurrent TCP sessions allowed from an individual source IP address or subnet.

```
[edit services ids rule rule-name term term-name then]
user@host# set session-limit by-source by-protocol tcp maximum number
```

- Configure the maximum number of TCP packets per second allowed from an individual source IP address or subnet.

```
[edit services ids rule rule-name term term-name then]
user@host# set session-limit by-source by-protocol tcp packets number
```

- Configure the maximum number of TCP connections per second allowed from an individual source IP address or subnet.

```
[edit services ids rule rule-name term term-name then]
user@host# set session-limit by-source by-protocol tcp rate number
```

- Configure the maximum number of concurrent TCP sessions allowed for an individual destination IP address or subnet.

```
[edit services ids rule rule-name term term-name then]
user@host# set session-limit by-destination by-protocol tcp maximum number
```

- Configure the maximum number of TCP connections per second allowed for an individual destination IP address or subnet.

```
[edit services ids rule rule-name term term-name then]
user@host# set session-limit by-destination by-protocol tcp rate number
```

- Configure the maximum number of TCP packets per second allowed for an individual destination IP address or subnet.

```
[edit services ids rule rule-name term term-name then]
user@host# set session-limit by-destination by-protocol tcp packets number
```

- Configure the closing of unestablished TCP connections and the delivery of a TCP RST to the end host to clear the TCP states on it when the open-timeout value at the [edit interfaces *interface-name* service-options] hierarchy level expires.

```
[edit services ids rule rule-name term term-name then]
user@host# set tcp-syn-defense
```

Configuring ICMP Fragmentation Protection

To protect against ICMP fragmentation attacks:

- Configure the identification and dropping of ICMP packets that are IP fragments.

```
[edit services ids rule rule-name term term-name then]
user@host# set icmp-fragment-check
```

Configuring ICMP Large Packet Protection

To protect against ICMP large packet attacks:

- Configure the identification and dropping of ICMP packets that are larger than 1024 bytes.

```
[edit services ids rule rule-name term term-name then]
user@host# set icmp-large-packet-check
```

Configuring IP Bad Options Protection

To protect against bad IPv4 options or IPv6 extension header attacks:

1. Configure the type of IPv4 options that the packet can include. If the packet includes an option that is not configured, then the packet is blocked. If the packet includes a configured option whose length is an illegal value, then the packet is dropped. Specifying any allows all options.

   ```
   [edit services ids rule rule-name term term-name then]
   user@host# set allow-ip-options [ip-options]
   ```

   The IPv4 options supported are any, loose-source-route, route-record, security, stream-id, strict-source-route, and timestamp.

   If you do not include the allow-ip-options statement in the IDS rule, packets with any type of IPv4 option are blocked.

2. Configure the type of IPv6 extension headers that the packet can include. If the packet includes an extension header that is not configured, then the packet is blocked. If the packet includes configured

extension headers that are incorrect, then the packet is dropped. Specifying `any` allows all extension headers.

```
[edit services ids rule rule-name term term-name then]
user@host# set allow-ipv6-extension-header extension-header
```

The IPv6 extension headers supported are `any`, `ah`, `dstopts`, `esp`, `fragment`, `hop-by-hop`, `mobility`, and `routing`.

If you do not include the `allow-ipv6-extension-header` statement in the IDS rule, packets with any type of extension header are dropped.

Configuring Land Attack Protection

To protect against land attacks:

- Configure the identification and dropping of SYN packets that have the same source and destination IP address or the same source and destination IP address and port.

```
[edit services ids rule rule-name term term-name then]
user@host# set land-attack-check (ip-only | ip-port)
```

To specify that the packets have the same source and destination IP address, use the `ip-only` option; to specify that the packets have the same source and destination IP address and port, use the `ip-port` option.

Configuring TCP SYN Fragment Protection

To protect against TCP SYN fragment attacks:

- Configure the identification and dropping of TCP SYN packets that are IP fragments:

```
[edit services ids rule rule-name term term-name then]
user@host# set tcp-syn-fragment-check
```

Configuring WinNuke Protection

To protect against WinNuke attacks:

- Configure the identification and dropping of TCP segments that are destined for port 139 and have the urgent (URG) flag set.

```
[edit services ids rule rule-name term term-name then]
user@host# set tcp-winnuke-check
```

Configuring the Service Set

To apply the IDS rule actions to a service set:

1. Assign the IDS rule to a service set that is on an MS-MPC.

   ```
   [edit services]
   user@host# set service-set service-set-name ids-rules rule-name
   ```

   If the service set is associated with an AMS interface, then the session limits you configure are applicable to each member interface.

2. Limit the packets that the IDS rule processes by configuring a stateful firewall rule (see "Configuring Stateful Firewall Rules" on page 550). The stateful firewall rule can identify either the traffic that should undergo IDS processing or the traffic that should skip IDS processing:

   - To allow IDS processing on the traffic that matches the stateful firewall rule, include `accept` at the [edit services stateful-firewall rule *rule-name* term *term-name* then] hierarchy level.

   - To skip IDS processing on the traffic that matches the stateful firewall rule, include `accept skip-ids` at the [edit services stateful-firewall rule *rule-name* term *term-name* then] hierarchy level.

3. Assign the stateful firewall rule to the service set.

   ```
   [edit services]
   user@host# set service-set service-set-name stateful-firewall-rules rule-name
   ```

Configuring Protection Against Header Anomaly Attacks

Protect against header anomaly attacks by using either of the following methods to enable a header integrity check, which drops any packets with header anomalies:

- Configure a stateful firewall rule, a NAT rule, or an IDS rule and apply it to the service set that is on an MS-MPC. A header integrity check is automatically enabled.

- Configure a header integrity check for the service set that is on an MS-MPC.

  ```
  [edit services]
  user@host# set service-set service-set-name service-set-options header-integrity-check enable-
  all
  ```

Configuring Logging of Network Attack Protection Packet Drops on an MS-MPC

To configure the logging of packet drops resulting from header integrity, suspicious packet pattern, and session limit checks performed by an MS-MPC:

1. Configure the logging of packet drops resulting from header integrity failures and suspicious packet patterns.

```
[edit services set service-set service-set-name syslog host hostname class]
user@host# set packet-logs
```

2. Configure the logging of packet drops resulting from session limit violations.

```
[edit services set service-set service-set-name syslog host hostname class]
user@host# set ids-log
```

SEE ALSO

Configuring Protection Against Network Attacks on an MS-MPC | **610**

7

PART

IPsec Tunnels

IPsec Overview | 623

IPsec Tunnels With Static Endpoints | 643

IPsec Tunnels With Dynamic Endpoints | 817

CHAPTER 29

IPsec Overview

IN THIS CHAPTER

- IPsec Overview | 623

IPsec Overview

IN THIS SECTION

- Understanding Junos VPN Site Secure | 623
- Authentication Algorithms | 627
- Encryption Algorithms | 628
- IPsec Protocols | 629
- IPsec Multipath Forwarding with UDP Encapsulation | 632
- Supported IPsec and IKE Standards | 634
- IPSec Terms and Acronyms | 636
- IPsec for ACX Series Overview | 640

Understanding Junos VPN Site Secure

IN THIS SECTION

- IPsec | 624
- Security Associations | 624
- IKE | 625

- Non-Support for NAT-T | 625
- Comparison of IPsec on ES PICs and Junos VPN Site Secure on Multiservices LIne Cards | 626

Junos VPN Site Secure is a suite of IPsec features supported on multiservices line cards (MS-DPC, MS-MPC, and MS-MIC), and was referred to as IPsec services in Junos releases earlier than 13.2. In Junos OS Release 13.2 and later, the term IPsec features is used exclusively to refer to the IPsec implementation on Adaptive Services and Encryption Services PICs. This topic provides you an overview of Junos VPN Site Secure, and has the following sections:

IPsec

The IPsec architecture provides a security suite for the IP version 4 (IPv4) and IP version 6 (IPv6) network layers. The suite provides such functionality as authentication of origin, data integrity, confidentiality, replay protection, and nonrepudiation of source. In addition to IPsec, the Junos OS also supports the Internet Key Exchange (IKE), which defines mechanisms for key generation and exchange, and manages security associations (SAs).

IPsec also defines a security association and key management framework that can be used with any network-layer protocol. The SA specifies what protection policy to apply to traffic between two IP-layer entities. IPsec provides secure tunnels between two peers.

Security Associations

To use IPsec security services, you create SAs between hosts. An SA is a simplex connection that enables two hosts to communicate with each other securely by means of IPsec. There are two types of SAs:

- Manual SAs require no negotiation; all values, including the keys, are static and specified in the configuration. Manual SAs statically define the security parameter index (SPI) values, algorithms, and keys to be used, and require matching configurations on both ends of the tunnel. Each peer must have the same configured options for communication to take place.

- Dynamic SAs require additional configuration. With dynamic SAs, you configure IKE first and then the SA. IKE creates dynamic security associations; it negotiates SAs for IPsec. The IKE configuration defines the algorithms and keys used to establish the secure IKE connection with the peer security gateway. This connection is then used to dynamically agree upon keys and other data used by the dynamic IPsec SA. The IKE SA is negotiated first and then used to protect the negotiations that determine the dynamic IPsec SAs.

IKE

IKE is a key management protocol that creates dynamic SAs; it negotiates SAs for IPsec. An IKE configuration defines the algorithms and keys used to establish a secure connection with a peer security gateway.

IKE performs the following tasks:

- Negotiates and manages IKE and IPsec parameters.
- Authenticates secure key exchange.
- Provides mutual peer authentication by means of shared secrets (not passwords) and public keys.
- Provides identity protection (in main mode).

Two versions of the IKE protocol (IKEv1 and IKEv2) are supported now. IKE negotiates security attributes and establishes shared secrets to form the bidirectional IKE SA. In IKE, inbound and outbound IPsec SAs are established and the IKE SA secures the exchanges. Starting with Junos OS Release 11.4, both IKEv1 and IKEv2 are supported by default on all M Series, MX Series, and T Series routers. IKE also generates keying material, provides Perfect Forward Secrecy, and exchanges identities.

Starting in Junos OS Release 18.2R1, you can configure an MX Series router with MS-MPCs or MS-MICs to act only as an IKE responder. In this responder-only mode, the MX Series router does not initiate IKE negotiations, it only responds to IKE negotiations initiated by the peer gateway. This might be required when inter-operating with other vendor's equipment, such as Cisco devices. Because the MX Series does not support the protocol and port values in the traffic selector, it cannot initiate an IPsec tunnel to another vendor's peer gateway that expects these values. By configuring the response-only mode on the MX Series, the MX can accept the traffic selector in the IKE negotiation initiated from the peer gateway.

Starting in Junos OS Release 18.2R1, you can configure the MX Series router with MS-MPCs or MS-MICs to send only the end-entity certificate for certificate-based IKE authentication instead of the full certificate chain. This avoids IKE fragmentation.

Starting with Junos OS Release 19.1R1, distinguished name support is added to the IKE identification (IKE ID) that is used for validation of VPN peer devices during IKE negotiation. The IKE ID received by an MX Series router from a remote peer can be an IPv4 or an IPv6 address, a hostname, a fully qualified domain name (FQDN), or a distinguished name (DN). The IKE ID sent by the remote peer needs to match what is expected by the MX Series router. Otherwise, IKE ID validation fails and the VPN is not established.

Non-Support for NAT-T

Before Junos OS Release 17.4R1, Network Address Translation-Traversal (NAT-T) is not supported for the Junos VPN Site Secure suite of IPsec features on the MX Series routers, and you must disable NAT-T on the MX Series router to avoid running unsupported NAT-T (see "Disabling NAT-T on MX Series Routers for Handling NAT with IPsec-Protected Packets" on page 811). NAT-T is a method for getting around IP address translation issues encountered when data protected by IPsec passes through a NAT device for address translation.

Comparison of IPsec on ES PICs and Junos VPN Site Secure on Multiservices LIne Cards

Table 24 on page 626 compares the top-level configuration of IPsec features on the ES PIC interfaces, and IPsec on the Adaptive Services PICs and Junos VPN Site Secure on Multiservices Line Cards .

Table 24: Statement Equivalents for ES and AS Interfaces

ES PIC Configuration	AS and MultiServices Line Cards Configuration

`[edit security ipsec]`
`proposal {...}`
	`[edit services ipsec-vpn ipsec]`
	`proposal {...}`

`[edit security ipsec]`
`policy {...}`
	`[edit services ipsec-vpn ipsec]`
	`policy {...}`

`[edit security ipsec]`
`security-association sa-dynamic {...}`
	`[edit services ipsec-vpn rule` *`rule-name`*`]`
	`term` *`term-name`* `match-conditions {...}`
	`then dynamic {...}]`

`[edit security ipsec]`
`security-association sa-manual {...}`
	`[edit services ipsec-vpn rule` *`rule-name`*`]`
	`term` *`term-name`* `match-conditions {...}`
	`then manual {...}]`

`[edit security ike]`
`proposal {...}`
	`[edit services ipsec-vpn ike]`
	`proposal {...}`

`[edit security ike]`
`policy {...}`
	`[edit services ipsec-vpn ike]`
	`policy {...}`

Table 24: Statement Equivalents for ES and AS Interfaces *(Continued)*

ES PIC Configuration	AS and MultiServices Line Cards Configuration

Not available	`[edit services ipsec-vpn]`
	`rule-set {...}`

Not available	`[edit services ipsec-vpn]`
	`service-set {...}`

`[edit interfaces es-`*`fpc/pic/port`*`]`
`tunnel source` *`address`*
	`[edit services ipsec-vpn service-set` *`set-name`* `ipsec-vpn local-gateway` *`address`*`]`

`[edit interfaces es-`*`fpc/pic/port`*`]`
`tunnel destination` *`address`*
	`[edit services ipsec-vpn rule` *`rule-name`*`]`
	`remote-gateway` *`address`*

NOTE: Although many of the same statements and properties are valid on both platforms (MultiServices and ES), the configurations are not interchangeable. You must commit a complete configuration for the PIC type that is installed in your router.

Authentication Algorithms

Authentication is the process of verifying the identity of the sender. Authentication algorithms use a shared key to verify the authenticity of the IPsec devices. The Junos OS uses the following authentication algorithms:

- Message Digest 5 (MD5) uses a one-way hash function to convert a message of arbitrary length to a fixed-length message digest of 128 bits. Because of the conversion process, it is mathematically infeasible to calculate the original message by computing it backwards from the resulting message digest. Likewise, a change to a single character in the message will cause it to generate a very different message digest number.

  To verify that the message has not been tampered with, the Junos OS compares the calculated message digest against a message digest that is decrypted with a shared key. The Junos OS uses the MD5 hashed message authentication code (HMAC) variant that provides an additional level of hashing. MD5 can be used with authentication header (AH), Encapsulating Security Payload (ESP), and Internet Key Exchange (IKE).

- Secure Hash Algorithm 1 (SHA-1) uses a stronger algorithm than MD5. SHA-1 takes a message of less than 264 bits in length and produces a 160-bit message digest. The large message digest ensures that the data has not been changed and that it originates from the correct source. The Junos OS uses the SHA-1 HMAC variant that provides an additional level of hashing. SHA-1 can be used with AH, ESP, and IKE.

- SHA-256, SHA-384, and SHA-512 (sometimes grouped under the name SHA-2) are variants of SHA-1 and use longer message digests. The Junos OS supports the SHA-256 version of SHA-2, which can process all versions of Advanced Encryption Standard (AES), Data Encryption Standard (DES), and Triple DES (3DES) encryption.

Encryption Algorithms

Encryption encodes data into a secure format so that it cannot be deciphered by unauthorized users. Like authentication algorithms, a shared key is used with encryption algorithms to verify the authenticity of the IPsec devices. The Junos OS uses the following encryption algorithms:

- Data Encryption Standard cipher-block chaining (DES-CBC) is a symmetric secret-key block algorithm. DES uses a key size of 64 bits, where 8 bits are used for error detection and the remaining 56 bits provide encryption. DES performs a series of simple logical operations on the shared key, including permutations and substitutions. CBC takes the first block of 64 bits of output from DES, combines this block with the second block, feeds this back into the DES algorithm, and repeats this process for all subsequent blocks.

- Triple DES-CBC (3DES-CBC) is an encryption algorithm that is similar to DES-CBC, but provides a much stronger encryption result because it uses three keys for 168-bit (3 x 56-bit) encryption. 3DES works by using the first key to encrypt the blocks, the second key to decrypt the blocks, and the third key to re-encrypt the blocks.

- Advanced Encryption Standard (AES) is a next-generation encryption method based on the Rijndael algorithm developed by Belgian cryptographers Dr. Joan Daemen and Dr. Vincent Rijmen. It uses a 128-bit block and three different key sizes (128, 192, and 256 bits). Depending on the key size, the algorithm performs a series of computations (10, 12, or 14 rounds) that include byte substitution, column mixing, row shifting, and key addition. The use of AES in conjunction with IPsec is defined in RFC 3602, *The AES-CBC Cipher Algorithm and Its Use with IPsec*.

- Starting In Junos OS Release 17.3R1, Advanced Encryption Standard in Galois/Counter Mode (AES-GCM) is supported for MS-MPCs and MS-MICs. However, in Junos FIPS mode, AES-GCM is not supported in Junos OS Release 17.3R1. Starting in Junos OS Release 17.4R1, AES-GCM is supported in Junos FIPS mode. AES-GCM is an authenticated encryption algorithm designed to provide both authentication and privacy. AES-GCM uses universal hashing over a binary Galois field to provide authenticated encryption and allows authenticated encryption at data rates of tens of Gbps.

SEE ALSO

Configuring IKE Proposals | 675

Configuring IPsec Proposals | 688

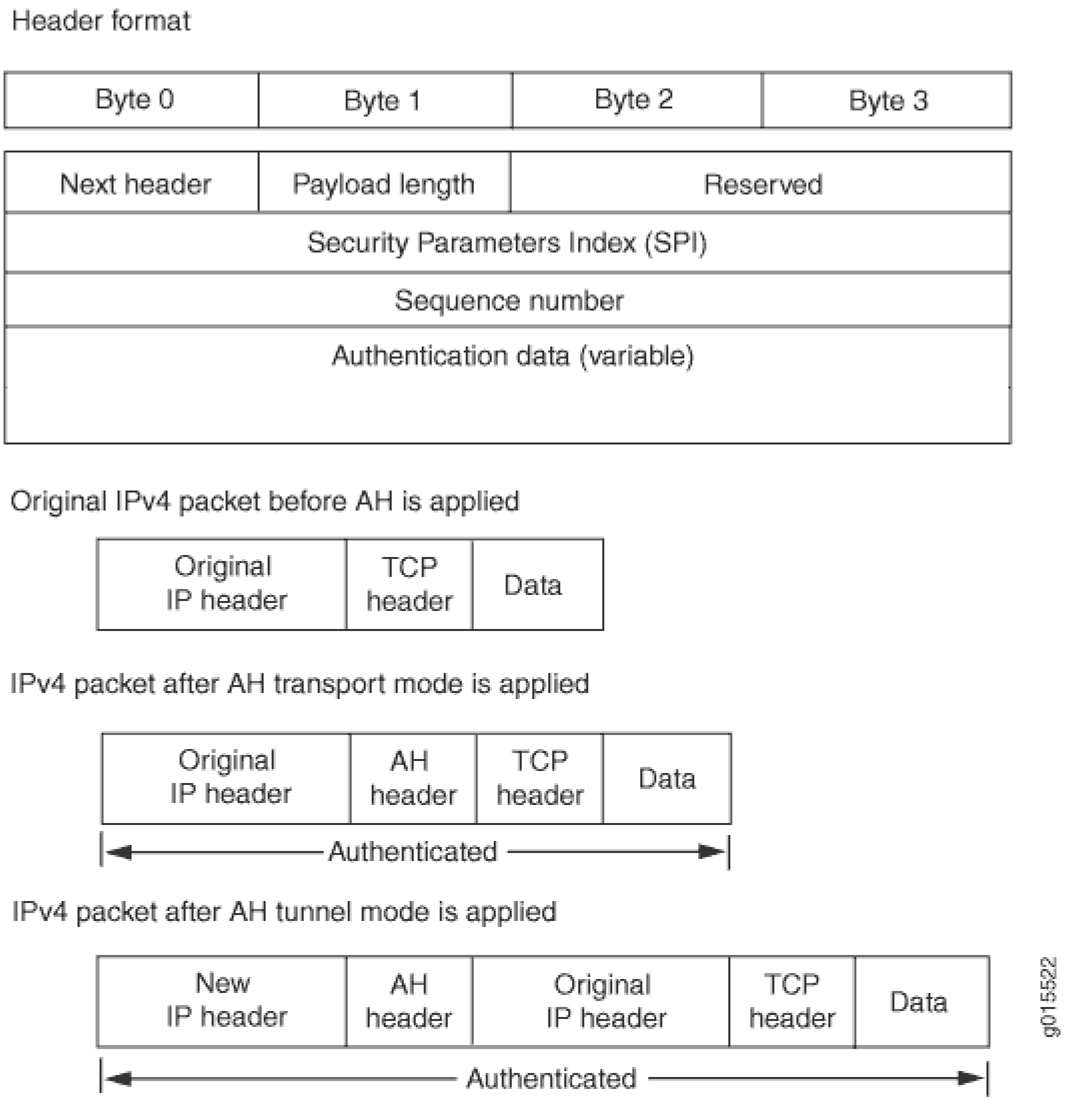
IPsec Protocols

IPsec protocols determine the type of authentication and encryption applied to packets that are secured by the router. The Junos OS supports the following IPsec protocols:

- AH—Defined in RFC 2402, AH provides connectionless integrity and data origin authentication for IPv4 and IPv6 packets. It also provides protection against replays. AH authenticates as much of the IP header as possible, as well as the upper-level protocol data. However, some IP header fields might change in transit. Because the value of these fields might not be predictable by the sender, they cannot be protected by AH. In an IP header, AH can be identified with a value of `51` in the `Protocol` field of an IPv4 packet and the `Next Header` field of an IPv6 packet. An example of the IPsec protection offered by AH is shown in Figure 33 on page 630 .

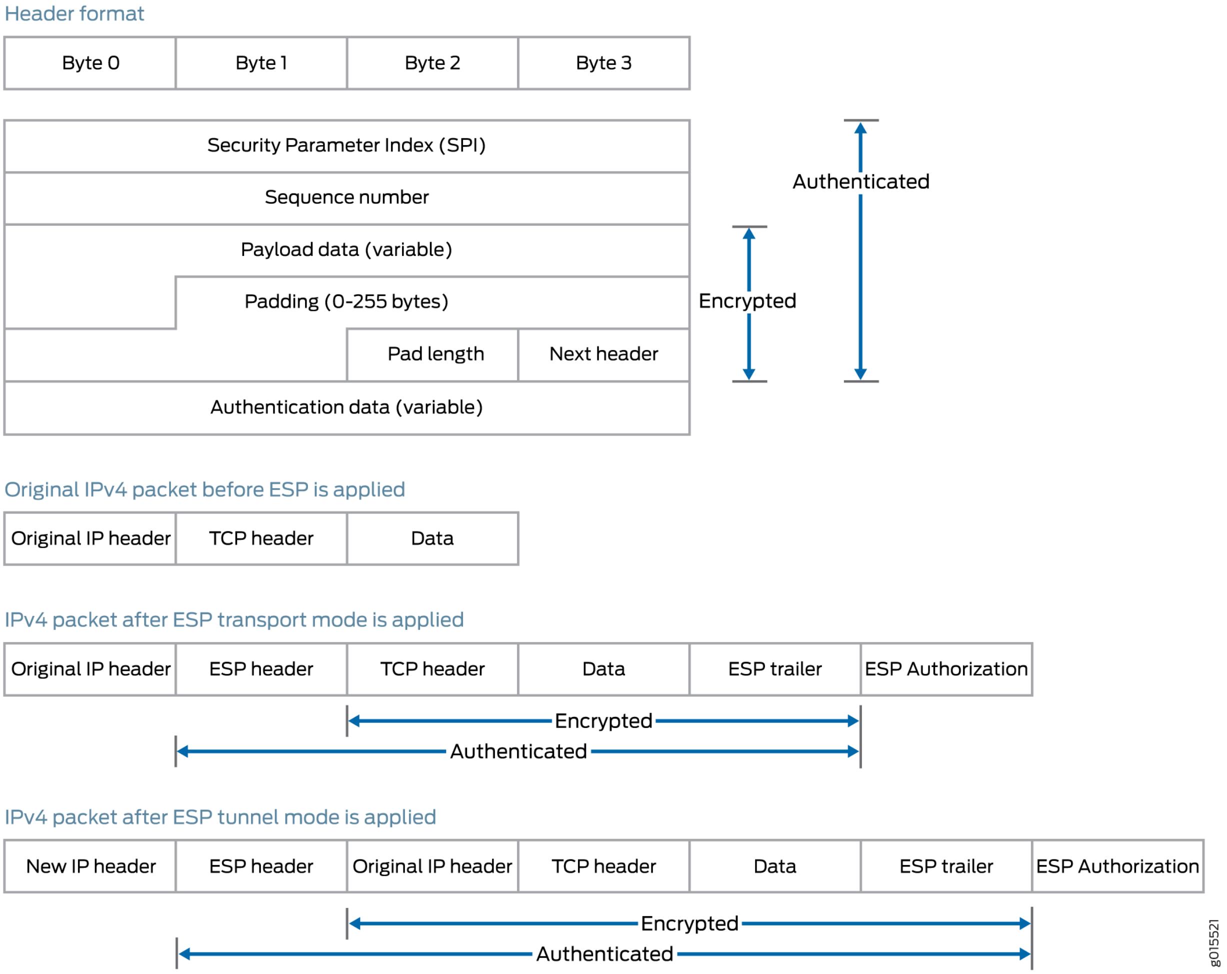
  NOTE: AH is not supported on the T Series, M120, and M320 routers.

Figure 33: AH Protocol



- ESP—Defined in RFC 2406, ESP can provide encryption and limited traffic flow confidentiality, or connectionless integrity, data origin authentication, and an anti-replay service. In an IP header, ESP can be identified a value of `50` in the `Protocol` field of an IPv4 packet and the `Next Header` field of an IPv6 packet. An example of the IPsec protection offered by ESP is shown in Figure 34 on page 631 .

Figure 34: ESP Protocol



- Bundle—When you compare AH with ESP, there are some benefits and shortcomings in both protocols. ESP provides a decent level of authentication and encryption, but does so only for part of the IP packet. Conversely, although AH does not provide encryption, it does provide authentication for the entire IP packet. Because of this, the Junos OS offers a third form of IPsec protocol called a protocol bundle. The bundle option offers a hybrid combination of AH authentication with ESP encryption.

SEE ALSO

Configuring IPsec Proposals | 688

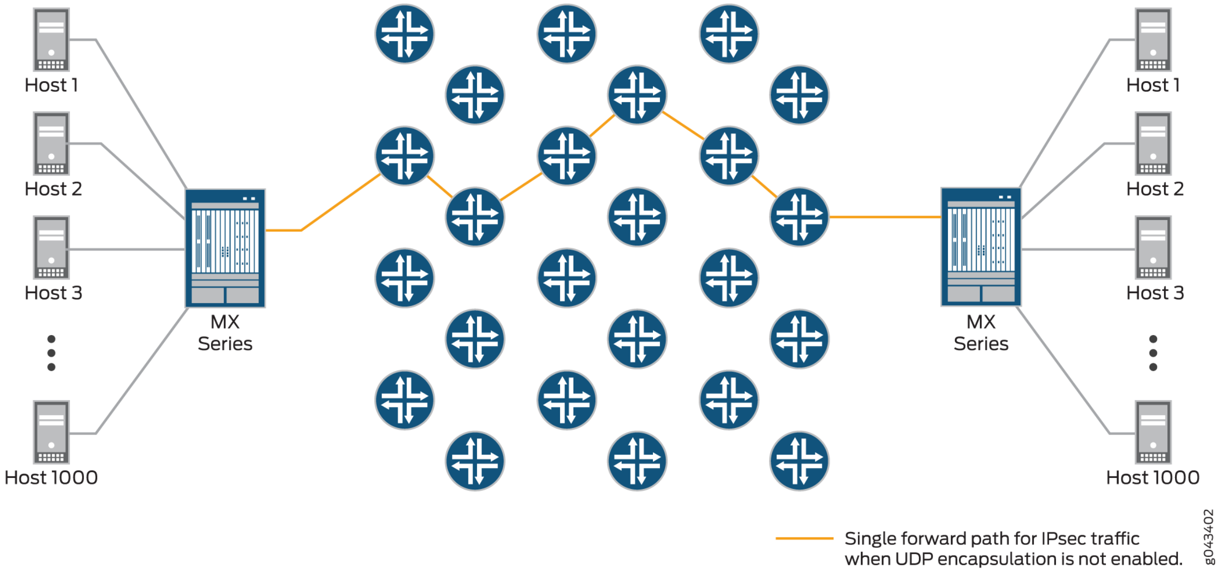
Configuring Security Associations | 645

protocol (IPsec)

IPsec Multipath Forwarding with UDP Encapsulation

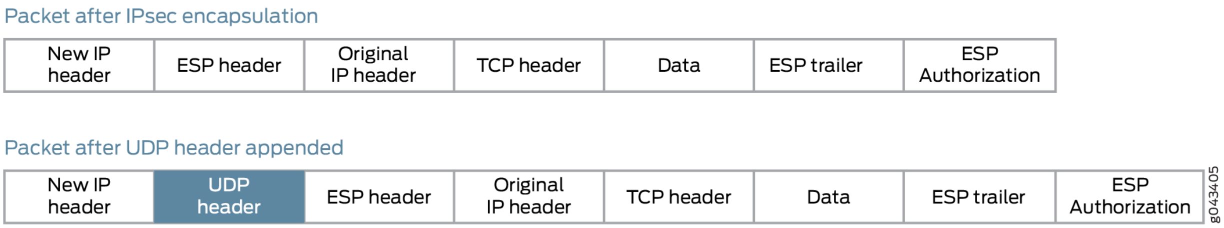
IPsec provides secure tunnels between two peers, and IPsec encapsulated packets have IP headers that contain tunnel endpoint IPs that do not change. This results in the selection of a single forwarding path between the peers, as shown in Figure 35 on page 632 . When IPsec traffic is flowing between data centers with thousands of hosts, this single path selection limits the throughput.

Figure 35: IPsec with One Forwarding Path



You can overcome this problem by enabling UDP encapsulation of the IPsec packets, which appends a UDP header after the ESP header, as shown in Figure 36 on page 632 . This provides Layer 3 and 4 information to the intermediate routers, and the IPsec packets are forwarded over multiple paths, as shown in Figure 37 on page 633 . You enable UDP encapsulation for the service set.

Figure 36: Appended UDP Header



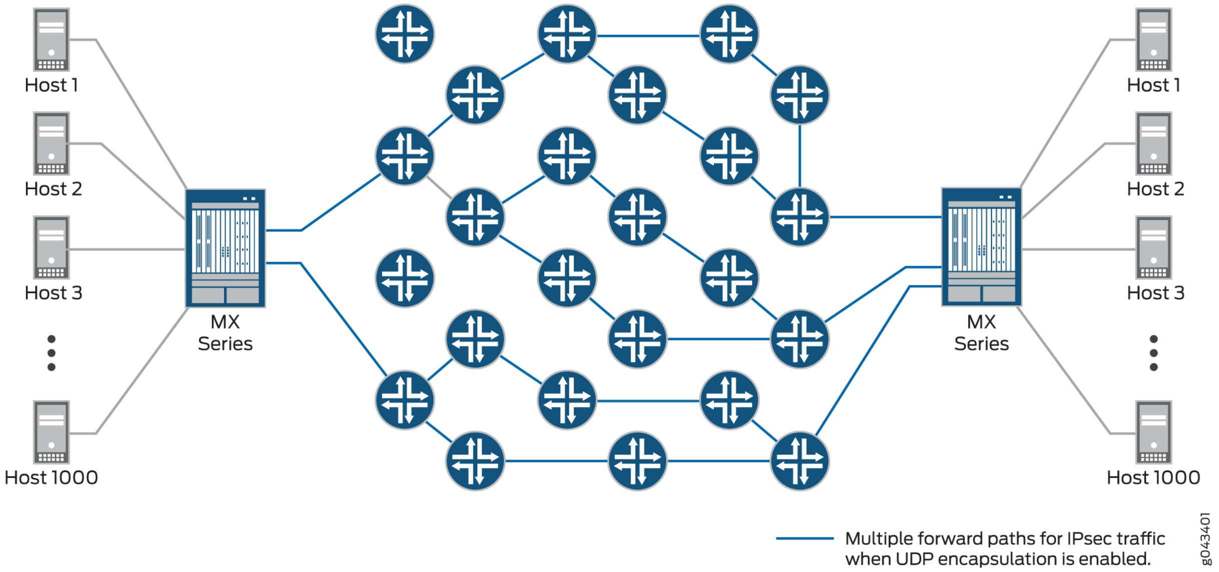
Packet after IPsec encapsulation

New IP header	ESP header	Original IP header	TCP header	Data	ESP trailer	ESP Authorization

Packet after UDP header appended

New IP header	UDP header	ESP header	Original IP header	TCP header	Data	ESP trailer	ESP Authorization

Figure 37: IPsec with Multiple Forwarding Paths



You can configure the UDP destination port or use the default value of 4565. You cannot configure 4500 as the destination port because it is a well-known port for NAT traversals.

Junos OS generates the source UDP port through a hash function that operates on the following data:

- Source IP address
- Destination IP address
- Transport protocol
- Transport source port
- Transport destination port
- A random number

Only the last two bytes of the resulting hash are used, so the internal IP header details are hidden.

When NAT-T is detected, only NAT-T UDP encapsulation occurs, not the UDP encapsulation for IPsec packets.

SEE ALSO

Configuring IPsec Service Sets | 734

Supported IPsec and IKE Standards

On routers equipped with one or more MS-MPCs, MS-MICs, or DPCs, the Canada and U.S. version of Junos OS substantially supports the following RFCs, which define standards for IP Security (IPsec) and Internet Key Exchange (IKE).

- RFC 2085, *HMAC-MD5 IP Authentication with Replay Prevention*
- RFC 2401, *Security Architecture for the Internet Protocol* (obsoleted by RFC 4301)
- RFC 2402, *IP Authentication Header* (obsoleted by RFC 4302)
- RFC 2403, *The Use of HMAC-MD5-96 within ESP and AH*
- RFC 2404, *The Use of HMAC-SHA-1-96 within ESP and AH* (obsoleted by RFC 4305)
- RFC 2405, *The ESP DES-CBC Cipher Algorithm With Explicit IV*
- RFC 2406, *IP Encapsulating Security Payload (ESP)* (obsoleted by RFC 4303 and RFC 4305)
- RFC 2407, *The Internet IP Security Domain of Interpretation for ISAKMP* (obsoleted by RFC 4306)
- RFC 2408, *Internet Security Association and Key Management Protocol (ISAKMP)* (obsoleted by RFC 4306)
- RFC 2409, *The Internet Key Exchange (IKE)* (obsoleted by RFC 4306)
- RFC 2410, *The NULL Encryption Algorithm and Its Use With IPsec*
- RFC 2451, *The ESP CBC-Mode Cipher Algorithms*
- RFC 2560, *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP*
- RFC 3193, *Securing L2TP using IPsec*
- RFC 3280, *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*
- RFC 3602, *The AES-CBC Cipher Algorithm and Its Use with IPsec*
- RFC 3948, *UDP Encapsulation of IPsec ESP Packets*
- RFC 4106, *The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP)*
- RFC 4210, *Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP)*
- RFC 4211, *Internet X.509 Public Key Infrastructure Certificate Request Message Format (CRMF)*
- RFC 4301, *Security Architecture for the Internet Protocol*

- RFC 4302, *IP Authentication Header*
- RFC 4303, *IP Encapsulating Security Payload (ESP)*
- RFC 4305, *Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH)*
- RFC 4306, *Internet Key Exchange (IKEv2) Protocol*
- RFC 4307, *Cryptographic Algorithms for Use in the Internet Key Exchange Version 2 (IKEv2)*
- RFC 4308, *Cryptographic Suites for IPsec*

  Only Suite VPN-A is supported in Junos OS.
- RFC 4754, *IKE and IKEv2 Authentication Using the Elliptic Curve Digital Signature Algorithm (ECDSA)*
- RFC 4835, *Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH)*
- RFC 5996, *Internet Key Exchange Protocol Version 2 (IKEv2)* (obsoleted by RFC 7296)
- RFC 7296, *Internet Key Exchange Protocol Version 2 (IKEv2)*
- RFC 8200, *Internet Protocol, Version 6 (IPv6) Specification*

Junos OS partially supports the following RFCs for IPsec and IKE:

- RFC 3526, *More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE)*
- RFC 5114, *Additional Diffie-Hellman Groups for Use with IETF Standards*
- RFC 5903, *Elliptic Curve Groups modulo a Prime (ECP Groups) for IKE and IKEv2*

The following RFCs and Internet draft do not define standards, but provide information about IPsec, IKE, and related technologies. The IETF classifies them as “Informational.”

- RFC 2104, *HMAC: Keyed-Hashing for Message Authentication*
- RFC 2412, *The OAKLEY Key Determination Protocol*
- RFC 3706, *A Traffic-Based Method of Detecting Dead Internet Key Exchange (IKE) Peers*
- Internet draft draft-eastlake-sha2-02.txt, *US Secure Hash Algorithms (SHA and HMAC-SHA)* (expires July 2006)

SEE ALSO

- Services Interfaces Overview for Routing Devices
- MX Series 5G Universal Routing Platform Interface Module Reference
- Accessing Standards Documents on the Internet

IPSec Terms and Acronyms

IN THIS SECTION

- Triple Data Encryption Standard (3DES) | **637**
- Adaptive Services PIC | **637**
- Advanced Encryption Standard (AES) | **637**
- authentication header (AH) | **637**
- certificate authority (CA) | **637**
- certificate revocation list (CRL) | **637**
- cipher block chaining (CBC) | **637**
- Data Encryption Standard (DES) | **638**
- digital certificate | **638**
- ES PIC | **638**
- Encapsulating Security Payload (ESP) | **638**
- Hashed Message Authentication Code (HMAC) | **638**
- Internet Key Exchange (IKE) | **638**
- Message Digest 5 (MD5) | **638**
- Perfect Forward Secrecy (PFS) | **638**
- public key infrastructure (PKI) | **639**
- registration authority (RA) | **639**
- Routing Engine | **639**
- security association (SA) | **639**
- Security Association Database (SADB) | **639**
- Secure Hash Algorithm 1 (SHA-1) | **639**
- Secure Hash Algorithm 2 (SHA-2) | **639**
- Security Policy Database (SPD) | **639**
- Security Parameter Index (SPI) | **640**
- Simple Certificate Enrollment Protocol (SCEP) | **640**

Triple Data Encryption Standard (3DES)

An enhanced DES algorithm that provides 168-bit encryption by processing data three times with three different keys.

Adaptive Services PIC

A next-generation Physical Interface Card (PIC) that provides IPsec services and other services, such as Network Address Translation (NAT) and stateful firewall, on M Series and T Series platforms.

Advanced Encryption Standard (AES)

A next-generation encryption method that is based on the Rijndael algorithm and uses a 128-bit block, three different key sizes (128, 192, and 256 bits), and multiple rounds of processing to encrypt data.

authentication header (AH)

A component of the IPsec protocol used to verify that the contents of a packet have not changed (data integrity), and to validate the identity of the sender (data source authentication). For more information about AH, see RFC 2402.

certificate authority (CA)

A trusted third-party organization that generates, enrolls, validates, and revokes digital certificates. The CA guarantees the identity of a user and issues public and private keys for message encryption and decryption.

certificate revocation list (CRL)

A list of digital certificates that have been invalidated before their expiration date, including the reasons for their revocation and the names of the entities that have issued them. A CRL prevents usage of digital certificates and signatures that have been compromised.

cipher block chaining (CBC)

A cryptographic method that encrypts blocks of ciphertext by using the encryption result of one block to encrypt the next block. Upon decryption, the validity of each block of ciphertext depends on the validity of all the preceding ciphertext blocks. For more information on how to use CBC with DES and ESP to provide confidentiality, see RFC 2405.

Data Encryption Standard (DES)

An encryption algorithm that encrypts and decrypts packet data by processing the data with a single shared key. DES operates in increments of 64-bit blocks and provides 56-bit encryption.

digital certificate

Electronic file that uses private and public key technology to verify the identity of a certificate creator and distribute keys to peers.

ES PIC

A PIC that provides first-generation encryption services and software support for IPsec on M Series and T Series platforms.

Encapsulating Security Payload (ESP)

A component of the IPsec protocol used to encrypt data in an IPv4 or IPv6 packet, provide data integrity, and ensure data source authentication. For more information about ESP, see RFC 2406.

Hashed Message Authentication Code (HMAC)

A mechanism for message authentication using cryptographic hash functions. HMAC can be used with any iterative cryptographic hash function, such as MD5 or SHA-1, in combination with a secret shared key. For more information on HMAC, see RFC 2104.

Internet Key Exchange (IKE)

Establishes shared security parameters for any hosts or routers using IPsec. IKE establishes the SAs for IPsec. For more information about IKE, see RFC 2407.

Message Digest 5 (MD5)

An authentication algorithm that takes a data message of arbitrary length and produces a 128-bit message digest. For more information, see RFC 1321.

Perfect Forward Secrecy (PFS)

Provides additional security by means of a Diffie-Hellman shared secret value. With PFS, if one key is compromised, previous and subsequent keys are secure because they are not derived from previous keys.

public key infrastructure (PKI)

A trust hierarchy that enables users of a public network to securely and privately exchange data through the use of public and private cryptographic key pairs that are obtained and shared with peers through a trusted authority.

registration authority (RA)

A trusted third-party organization that acts on behalf of a CA to guarantee the identity of a user.

Routing Engine

A PCI-based architectural portion of a Junos OS-based router that handles the routing protocol process, the interface process, some of the chassis components, system management, and user access.

security association (SA)

Specifications that must be agreed upon between two network devices before IKE or IPsec are allowed to function. SAs primarily specify protocol, authentication, and encryption options.

Security Association Database (SADB)

A database where all SAs are stored, monitored, and processed by IPsec.

Secure Hash Algorithm 1 (SHA-1)

An authentication algorithm that takes a data message of less than 264 bits in length and produces a 160-bit message digest. For more information on SHA-1, see RFC 3174.

Secure Hash Algorithm 2 (SHA-2)

A successor to the SHA-1 authentication algorithm that includes a group of SHA-1 variants (SHA-224, SHA-256, SHA-384, and SHA-512). SHA-2 algorithms use larger hash sizes and are designed to work with enhanced encryption algorithms such as AES.

Security Policy Database (SPD)

A database that works with the SADB to ensure maximum packet security. For inbound packets, IPsec checks the SPD to verify if the incoming packet matches the security configured for a particular policy. For outbound packets, IPsec checks the SPD to see if the packet needs to be secured.

Security Parameter Index (SPI)

An identifier that is used to uniquely identify an SA at a network host or router.

Simple Certificate Enrollment Protocol (SCEP)

A protocol that supports CA and registration authority (RA) public key distribution, certificate enrollment, certificate revocation, certificate queries, and certificate revocation list (CRL) queries.

IPsec for ACX Series Overview

IN THIS SECTION

- IPsec | 640
- Security Associations | 641
- IKE | 641

The Juniper Networks Junos operating system (Junos OS) supports IPsec. This topic includes the following sections, which provide background information about configuring IPsec on ACX Series Universal Metro Routers.

NOTE: IPsec is supported only on the ACX1100 AC-powered router and ACX500 routers. Service chaining (GRE, NAT, and IPSec) on ACX1100-AC and ACX500 routers is not supported.

NOTE: ACX5048 and ACX5096 routers do not support IPsec configurations.

For a list of the IPsec and IKE standards supported by Junos OS, see the *Junos OS Hierarchy and RFC Reference*.

IPsec

The IPsec architecture provides a security suite for the IP version 4 (IPv4) network layer. The suite provides functionality such as authentication of origin, data integrity, confidentiality, replay protection, and nonrepudiation of source. In addition to IPsec, Junos OS also supports the Internet Key Exchange (IKE), which defines mechanisms for key generation and exchange, and manages security associations.

IPsec also defines a security association and key management framework that can be used with any transport layer protocol. The security association specifies what protection policy to apply to traffic between two IP-layer entities. IPsec provides secure tunnels between two peers.

Security Associations

To use IPsec security services, you create security associations between hosts. A security association is a simplex connection that allows two hosts to communicate with each other securely by means of IPsec. There are two types of security associations:

- Manual security associations require no negotiation; all values, including the keys, are static and specified in the configuration. Manual security associations statically define the security parameter index (SPI) values, algorithms, and keys to be used, and require matching configurations on both ends of the tunnel. Each peer must have the same configured options for communication to take place.

- Dynamic security associations require additional configuration. With dynamic security associations, you configure IKE first and then the security association. IKE creates dynamic security associations; it negotiates security associations for IPsec. The IKE configuration defines the algorithms and keys used to establish the secure IKE connection with the peer security gateway. This connection is then used to dynamically agree upon keys and other data used by the dynamic IPsec security association. The IKE security association is negotiated first and then used to protect the negotiations that determine the dynamic IPsec security associations.

IKE

IKE is a key management protocol that creates dynamic security associations; it negotiates security associations for IPsec. An IKE configuration defines the algorithms and keys used to establish a secure connection with a peer security gateway.

IKE performs the following tasks:

- Negotiates and manages IKE and IPsec parameters.

- Authenticates secure key exchange.

- Provides mutual peer authentication by means of shared secrets (not passwords) and public keys.

- Provides identity protection (in main mode).

SEE ALSO

Enabling Inline Services Interface on ACX Series | 124

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
18.2R1	Starting in Junos OS Release 18.2R1, you can configure an MX Series router with MS-MPCs or MS-MICs to act only as an IKE responder.
18.2R1	Starting in Junos OS Release 18.2R1, you can configure the MX Series router with MS-MPCs or MS-MICs to send only the end-entity certificate for certificate-based IKE authentication instead of the full certificate chain.
17.4R1	Starting in Junos OS Release 17.4R1, AES-GCM is supported in Junos FIPS mode.
17.3R1	Starting In Junos OS Release 17.3R1, Advanced Encryption Standard in Galois/Counter Mode (AES-GCM) is supported for MS-MPCs and MS-MICs.

CHAPTER 30

IPsec Tunnels With Static Endpoints

IN THIS CHAPTER

- Minimum Security Association Configurations | 643
- Configuring Security Associations | 645
- Manual Security Associations | 653
- Dynamic Security Associations | 675
- IPsec Rules and Rulesets | 699
- Service Sets for Static Endpoint IPsec Tunnels | 733

Minimum Security Association Configurations

IN THIS SECTION

- Minimum Manual SA Configuration | 643
- Minimum Dynamic SA Configuration | 644

The following sections show the minimum configurations necessary to set up security associations (SAs) for IPsec services:

Minimum Manual SA Configuration

To define a manual SA configuration, you must include at least the following statements at the `[edit services ipsec-vpn rule rule-name term term-name then manual]` hierarchy level:

```
[edit services ipsec-vpn rule rule-name term term-name then manual]
direction (inbound | outbound | bidirectional) {
    authentication {
        algorithm (hmac-md5-96 | hmac-sha1-96);
        key (ascii-text key | hexadecimal key);
    }
    encryption {
        algorithm algorithm;
        key (ascii-text key | hexadecimal key);
    }
    protocol (ah | esp | bundle);
    spi spi-value;
}
```

Minimum Dynamic SA Configuration

To define a dynamic SA configuration, you must include at least the following statements at the `[edit services ipsec-vpn]` hierarchy level:

```
[edit services ipsec-vpn]
ike {
    proposal proposal-name {
        authentication-algorithm (md5 | sha1 | sha-256);
        authentication-method pre-shared-keys;
        dh-group (group1 | group2 | group5 |group14 | group15 | group16 | group19 | group20 |
group24);
        encryption-algorithm algorithm;
    }
    policy policy-name {
        proposals [ ike-proposal-names ];
        pre-shared-key (ascii-text key | hexadecimal key);
        version (1 | 2);
        mode (aggressive | main);
    }
}
ipsec {
    policy policy-name {
        proposals [ ipsec-proposal-names ];
    }
    proposal proposal-name {
        authentication-algorithm (hmac-md5-96 | hmac-sha1-96);
        encryption-algorithm algorithm;
        protocol (ah | esp | bundle);
    }
}
```

NOTE:

- Starting with Junos OS Release 11.4, both IKEv1 and IKEv2 are supported by default on all M Series, MX Series, and T Series routers. The `version` statement at the `[edit services ipsec-vpn ike policy *name*]` hierarchy level allows you to configure the specific IKE version to be supported.
- The `mode` statement at the `[edit services ipsec-vpn ike policy *name*]` hierarchy level is required only if the `version` option is set to **1**.

You must also include the `ipsec-policy` statement at the `[edit services ipsec-vpn rule *rule-name* term *term-name* then dynamic]` hierarchy level.

RELATED DOCUMENTATION

Understanding Junos VPN Site Secure | **623**

Configuring Security Associations | **645**

Configuring IKE Proposals | **675**

Configuring IKE Policies | **681**

Configuring IPsec Proposals | **688**

Configuring IPsec Policies | **694**

Configuring Security Associations

IN THIS SECTION

- Configuring Manual Security Associations | **646**
- Configuring Dynamic Security Associations | **651**
- Clearing Security Associations | **652**

To use IPsec services, you create a security association (SA) between hosts. An SA is a simplex connection that enables two hosts to communicate with each other securely using IPsec.

NOTE: Both OSPFv2 and OSPFv3 support IPsec authentication. However, dynamic or tunnel mode IPsec SAs are not supported for OSPFv3. If you add SAs into OSPFv3 by including the `ipsec-sa` statement at the `[edit protocols ospf3 area` *`area-number`* `interface` *`interface-name`*`]` hierarchy level, your configuration commit fails. For more information about OSPF authentication and other OSPF properties, see the Junos OS Routing Protocols Library.

You can configure two types of SAs:

- Manual—Requires no negotiation; all values, including the keys, are static and specified in the configuration. As a result, each peer must have the same configured options for communication to take place.

- Dynamic—Specifies proposals to be negotiated with the tunnel peer. The keys are generated as part of the negotiation and therefore do not need to be specified in the configuration. The dynamic SA includes one or more `proposal` statements that prioritizes a list of protocols and algorithms to be negotiated with the peer.

This section includes the following topics:

Configuring Manual Security Associations

IN THIS SECTION

- Configuring the Direction for IPsec Processing | 647
- Configuring the Protocol for a Manual IPsec SA | 648
- Configuring the Security Parameter Index | 649
- Configuring the Auxiliary Security Parameter Index | 649
- Configuring Authentication for a Manual IPsec SA | 650
- Configuring Encryption for a Manual IPsec SA | 650

Manual SAs require no negotiation; all values, including the keys, are static and specified in the configuration. As a result, each peer must have the same configured options for communication to take place. Manual SAs are best suited for small, static networks where the distribution, maintenance, and tracking of keys are not difficult.

To configure a manual IPsec security association, include the following statements at the `[edit services ipsec-vpn rule rule-name term term-name then manual]` hierarchy level:

```
[edit services ipsec-vpn rule rule-name term term-name then manual]
direction (inbound | outbound | bidirectional) {
    authentication {
        algorithm (hmac-md5-96 | hmac-sha-256-128 | hmac-sha1-96);
        key (ascii-text key | hexadecimal key);
    }
    auxiliary-spi auxiliary-spi-value;
    encryption {
        algorithm (3des-cbc | aes-128-cbc | aes-192-cbc | aes-256-cbc | des-cbc);;
        key (ascii-text key | hexadecimal key);
    }
    protocol (ah | esp | bundle);
    spi spi-value;
}
```

To configure manual SA statements, do the following:

Configuring the Direction for IPsec Processing

The `direction` statement specifies inbound or outbound IPsec processing. If you want to define different algorithms, keys, or security parameter index (SPI) values for each direction, you configure the `inbound` and `outbound` options. If you want the same attributes in both directions, use the `bidirectional` option.

To configure the direction of IPsec processing, include the `direction` statement at the `[edit services ipsec-vpn rule rule-name term term-name then manual]` hierarchy level:

```
[edit services ipsec-vpn rule rule-name term term-name then manual]
    direction (inbound | outbound | bidirectional) {
    ...
}
```

The following two examples illustrate this:

- Example: Using Different Configuration for the Inbound and Outbound Directions

  Define different algorithms, keys, and security parameter index values for each direction:

  ```
  [edit services ipsec-vpn rule rule-name term term-name then manual]
  direction bidirectional {
  ```

```
    protocol ah;
    spi 20001;
    authentication {
        algorithm hmac-md5-96;
        key ascii-text 123456789012abcd;
    }
}
direction outbound {
    protocol esp;
    spi 24576;
    encryption {
        algorithm 3des-cbc;
        key ascii-text 123456789012345678901234567890abcd;
    }
}
```

- Example: Using the Same Configuration for the Inbound and Outbound Directions

  Define one set of algorithms, keys, and security parameter index values that is valid in both directions:

```
[edit services ipsec-vpn rule rule-name term term-name then manual]
direction bidirectional {
    protocol ah;
    spi 20001;
    authentication {
        algorithm hmac-md5-96;
        key ascii-text 123456789012abcd;
    }
}
```

Configuring the Protocol for a Manual IPsec SA

IPsec uses two protocols to protect IP traffic: Encapsulating Security Payload (ESP) and authentication header (AH). The AH protocol is used for strong authentication. A third option, `bundle`, uses AH authentication and ESP encryption; it does not use ESP authentication because AH provides stronger authentication of IP packets.

To configure the IPsec protocol, include the `protocol` statement and specify the `ah`, `esp`, or `bundle` option at the `[edit services ipsec-vpn rule` *`rule-name`* `term` *`term-name`* `then manual direction` *`direction`*`]` hierarchy level:

```
[edit services ipsec-vpn rule rule-name term term-name then manual direction direction]
protocol (ah | bundle | esp);
```

Configuring the Security Parameter Index

An SPI is an arbitrary value that uniquely identifies which SA to use at the receiving host. The sending host uses the SPI to identify and select which SA to use to secure every packet. The receiving host uses the SPI to identify and select the encryption algorithm and key used to decrypt packets.

> **NOTE**: Each manual SA must have a unique SPI and protocol combination. Use the auxiliary SPI when you configure the `protocol` statement to use the `bundle` option.

To configure the SPI, include the `spi` statement and specify a value (from 256 through 16,639) at the `[edit services ipsec-vpn rule` *`rule-name`* `term` *`term-name`* `then manual direction` *`direction`*`]` hierarchy level:

```
[edit services ipsec-vpn rule rule-name term term-name then manual direction direction]
spi spi-value;
```

Configuring the Auxiliary Security Parameter Index

Use the auxiliary SPI when you configure the `protocol` statement to use the `bundle` option.

> **NOTE**: Each manual SA must have a unique SPI and protocol combination.

To configure the auxiliary SPI, include the `auxiliary-spi` statement and specify a value (from 256 through 16,639) at the `[edit services ipsec-vpn rule` *`rule-name`* `term` *`term-name`* `then manual direction` *`direction`*`]` hierarchy level:

```
[edit services ipsec-vpn rule rule-name term term-name then manual direction direction]
auxiliary-spi auxiliary-spi-value;
```

Configuring Authentication for a Manual IPsec SA

To configure an authentication algorithm, include the `authentication` statement and specify an authentication algorithm and a key at the `[edit services ipsec-vpn rule rule-name term term-name then manual direction direction]` hierarchy level:

```
[edit services ipsec-vpn rule rule-name term term-name then manual direction direction]
    authentication {
    algorithm (hmac-md5-96 | hmac-sha1-96 | hmac-sha-256-128)
    key (ascii-text key | hexadecimal key);
}
```

The algorithm can be one of the following:

- `hmac-md5-96`—Hash algorithm that authenticates packet data. It produces a 128-bit authenticator value and a 96-bit digest.
- `hmac-sha1-96`—Hash algorithm that authenticates packet data. It produces a 160-bit authenticator value and a 96-bit digest.
- `hmac-sha-256-128`—Hash algorithm that authenticates packet data. It produces a 256-bit authenticator value 256-bit digest, truncated to 128 bits.

The key can be one of the following:

- `ascii-text`—ASCII text key. With the `hmac-md5-96` option, the key contains 16 ASCII characters. With the `hmac-sha1-96` option, the key contains 20 ASCII characters.
- `hexadecimal`—Hexadecimal key. With the `hmac-md5-96` option, the key contains 32 hexadecimal characters. With the `hmac-sha1-96` option, the key contains 40 hexadecimal characters.

Configuring Encryption for a Manual IPsec SA

To configure IPsec encryption, include the `encryption` statement and specify an algorithm and key at the `[edit services ipsec-vpn rule rule-name term term-name then manual direction direction]` hierarchy level:

```
[edit services ipsec-vpn rule rule-name term term-name then manual direction direction]
    encryption {
    algorithm algorithm;
    key (ascii-text key | hexadecimal key);
}
```

The algorithm can be one of the following:

- des-cbc—Encryption algorithm that has a block size of 8 bytes; its key size is 64 bits long.

- 3des-cbc—Encryption algorithm that has a block size of 24 bytes; its key size is 192 bits long.

- aes-128-cbc—Advanced Encryption Standard (AES) 128-bit encryption algorithm.

- aes-192-cbc—Advanced Encryption Standard (AES) 192-bit encryption algorithm.

- aes-256-cbc—Advanced Encryption Standard (AES) 256-bit encryption algorithm.

NOTE: For a list of Data Encryption Standard (DES) encryption algorithm weak and semiweak keys, see RFC 2409, *The Internet Key Exchange (IKE)*. The AES encryption algorithms use a software implementation that has much lower throughput, so DES remains the recommended option. For reference information on AES encryption, see RFC 3602, *The AES-CBC Cipher Algorithm and Its Use with IPsec*.

For 3des-cbc, the first 8 bytes should differ from the second 8 bytes, and the second 8 bytes should be the same as the third 8 bytes.

If you configure an authentication proposal but do not include the encryption statement, the result is NULL encryption. Certain applications expect this result. If you configure no specific authentication or encryption values, the Junos OS uses the default values of sha1 for the authentication and 3des-cbc for the encryption.

The key can be one of the following:

- ascii-text—ASCII text key. With the des-cbc option, the key contains 8 ASCII characters. With the 3des-cbc option, the key contains 24 ASCII characters.

- hexadecimal—Hexadecimal key. With the des-cbc option, the key contains 16 hexadecimal characters. With the 3des-cbc option, the key contains 48 hexadecimal characters.

NOTE: You cannot configure encryption when you use the AH protocol.

Configuring Dynamic Security Associations

You configure dynamic SAs with a set of proposals that are negotiated by the security gateways. The keys are generated as part of the negotiation and therefore do not need to be specified in the configuration. The dynamic SA includes one or more proposals, which allow you to prioritize a list of protocols and algorithms to be negotiated with the peer.

To enable a dynamic SA, follow these steps:

1. Configure Internet Key Exchange (IKE) proposals and IKE policies associated with these proposals.

2. Configure IPsec proposals and an IPsec policy associated with these proposals.

3. Associate an SA with an IPsec policy by configuring the `dynamic` statement.

To configure a dynamic SA, include the `dynamic` statement and specify an IPsec policy name at the `[edit services ipsec-vpn rule` *`rule-name`* `term` *`term-name`* `then]` hierarchy level. The `ike-policy` statement is optional unless you use the preshared key authentication method.

```
[edit services ipsec-vpn rule rule-name term term-name then]
dynamic {
    ike-policy policy-name;
    ipsec-policy policy-name;
}
```

NOTE: If you want to establish a dynamic SA, the attributes in at least one configured IPsec and IKE proposal must match those of its peer.

Clearing Security Associations

You can set up the router software to clear IKE or IPsec SAs automatically when the corresponding services PIC restarts or is taken offline. To configure this property, include the `clear-ike-sas-on-pic-restart` or `clear-ipsec-sas-on-pic-restart` statement at the `[edit services ipsec-vpn]` hierarchy level:

```
[edit services ipsec-vpn]
clear-ike-sas-on-pic-restart;
clear-ipsec-sas-on-pic-restart;
```

After you add this statement to the configuration, all the IKE or IPsec SAs corresponding to the tunnels in the PIC will be cleared when the PIC restarts or goes offline.

Starting in Junos OS Release 17.2R1, you can enable the cleanup of IKE triggers and IKE and IPsec SAs when an IPsec tunnel's local gateway IP address goes down or the MS-MIC or MS-MPC being used in the tunnel's service set goes down. This reduces dropped traffic and unnecessary IKE triggers. To enable this feature, include the `gw-interface` statement at the `[edit services service set` *`service-set-name`* `ipsec-vpn-options local-gateway` *`address`*`]` hierarchy level. If the local gateway IP address for an IPsec tunnel's service set goes down or the MS-MIC or MS-MPC that is being used in the service set goes down, the service set no longer sends IKE triggers.

In addition, when the local gateway IP address goes down, the IKE and IPsec SAs are cleared for next-hop service sets, and go to the Not Installed state for interface-style service sets. The SAs that have the Not Installed state are deleted when the local gateway IP address comes back up. If the local gateway IP address that goes down for a next-hop service set is for the responder peer, then you need to clear the IKE and IPsec SAs on the initiator peer so that the IPsec tunnel comes back up once the local gateway IP address comes back up. You can either manually clear the IKE and IPsec SAs on the initiator peer or enable dead peer detection on the initiator peer.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
17.2R1	Starting in Junos OS Release 17.2R1, you can enable the cleanup of IKE triggers and IKE and IPsec SAs when an IPsec tunnel's local gateway IP address goes down or the MS-MIC or MS-MPC being used in the tunnel's service set goes down.

RELATED DOCUMENTATION

Configuring IPsec Policies | 694

Configuring IPsec Proposals | 688

Configuring IKE Policies | 681

Configuring IKE Proposals | 675

Manual Security Associations

IN THIS SECTION

- Example: Configuring Manual SAs | 653

Example: Configuring Manual SAs

IN THIS SECTION

- Requirements | 654

- Overview and Topology | 654
- Configuration | 655
- Verification | 671

This example shows how to create an IPsec tunnel by using manual security associations (SAs), and contains the following sections:

Requirements

This example uses the following hardware and software components:

- Four M Series, MX Series, or T Series routers with multiservices interfaces installed in them.
- Junos OS Release 9.4 and later.

No special configuration beyond device initialization is required before you can configure this feature.

Overview and Topology

IN THIS SECTION

- Topology | 655

A security association (SA) is a simplex connection that enables two hosts to securely communicate with each other by means of IPsec. There are two types of SAs: manual SA and dynamic SA. This example explains a manual SA configuration.

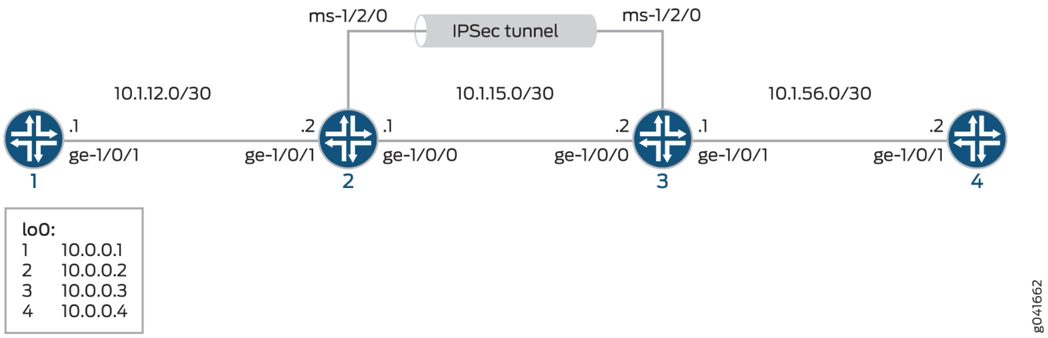
Manual SAs require no negotiation; all values, including the keys, are static and specified in the configuration. Manual SAs use statically defined security parameter index (SPI) values, algorithms, and keys, and require matching configurations on both ends of the tunnel. Each peer must have the same configured options for communication to take place.

Manual SAs are best suited for small, static networks where the distribution, maintenance, and tracking of keys are not difficult.

Topology

Figure 38 on page 655 shows an IPsec topology that contains a group of four routers: Routers 1, 2, 3, and 4.

Figure 38: Manual SA Topology



Routers 2 and 3 establish an IPsec tunnel by using a multiservices PIC and manual SA settings. Routers 1 and 4 provide basic connectivity and are used to verify that the IPsec tunnel is operational.

Configuration

IN THIS SECTION

- Configuring Router 1 | 656
- Configuring Router 2 | 658
- Configuring Router 3 | 664
- Configuring Router 4 | 669

This example uses four routers, and involves the following configurations:

- Routers 1 and 4 are configured for basic OSPF connectivity with Routers 2 and 3 respectively.

- Routers 2 and 3 are configured for OSPF connectivity with Routers 1 and 4 respectively. Routers 2 and 3 are also configured to create an IPsec tunnel by using manual SAs between these two routers. To direct traffic to the IPsec tunnel through the multiservices interface, next-hop style service sets are configured on Routers 2 and 3, and the multiservices interfaces that are configured as the IPsec inside interface are added to the OSPF configuration on the respective routers.

NOTE: The interface types shown in this example are for indicative purpose only. For example, you can use `so-` interfaces instead of `ge-` and `sp-` instead of `ms-`.

This section contains:

Configuring Router 1

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI, at the [edit] hierarchy level, of Router 1.

```
set interfaces ge-1/0/1 description "to R2 ge-1/0/1"
set interfaces ge-1/0/1 unit 0 family inet address 10.1.12.1/30
set interfaces lo0 unit 0 family inet address 10.0.0.1/32
set protocols ospf area 0.0.0.0 interface ge-1/0/1.0
set protocols ospf area 0.0.0.0 interface lo0.0
set routing-options router-id 10.0.0.1
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure Router 1 for OSPF connectivity with Router 2:

1. Configure an Ethernet interface and loopback interface.

```
[edit interfaces]
user@router1# set ge-1/0/1 description "to R2 ge-1/0/1"
user@router1# set ge-1/0/1 unit 0 family inet address 10.1.12.1/30
user@router1# set lo0 unit 0 family inet address 10.0.0.1/32
```

2. Specify the OSPF area and associate the interfaces with the OSPF area.

```
[edit protocols]
user@router1# set ospf area 0.0.0.0 interface ge-1/0/1.0
user@router1# set ospf area 0.0.0.0 interface lo0.0
```

3. Configure the router ID.

```
[edit routing-options]
user@router1# set router-id 10.0.0.1
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols ospf`, and `show routing-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration

```
user@router1# show interfaces
interfaces {
    ...
    ge-1/0/1 {
        description "to R2 ge-1/0/1";
        unit 0 {
            family inet {
                address 10.1.12.1/30;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.1/32;
            }
        }
    }
    ...
}
```

```
user@router1# show protocols ospf
ospf {
    area 0.0.0.0 {
        interface ge-1/0/1.0;
        interface lo0.0;
    }
}
```

```
user@router1# show routing-options
routing-options  {
    router-id 10.0.0.1;
}
```

Configuring Router 2

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI, at the [edit] hierarchy level, of Router 2.

Configuring Interfaces and OSPF Connectivity (with Router 1 and Router 3) on Router 2

```
set interfaces ge-1/0/0 unit 0 description "to R3 ge-1/0/0"
set interfaces ge-1/0/0 unit 0 family inet address 10.1.15.1/30
set interfaces ge-1/0/1 unit 0 description "to R1 ge-1/0/0"
set interfaces ge-1/0/1 unit 0 family inet address 10.1.12.2/30
set interfaces ms-1/2/0 unit 0 family inet
set interfaces ms-1/2/0 unit 1 family inet
set interfaces ms-1/2/0 unit 1 service-domain inside
set interfaces ms-1/2/0 unit 2 family inet
set interfaces ms-1/2/0 unit 2 service-domain outside
set interfaces lo0 unit 0 family inet address 10.0.0.2/32
set protocols ospf area 0.0.0.0 interface ge-1/0/1.0
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface ms-1/2/0.1
set services ipsec-vpn rule demo-rule-r1-manual-sa term demo-term-manual-sa then remote-gateway
10.1.15.2
set services ipsec-vpn rule demo-rule-r1-manual-sa term demo-term-manual-sa then manual
direction bidirectional protocol esp
set services ipsec-vpn rule demo-rule-r1-manual-sa term demo-term-manual-sa then manual
direction bidirectional spi 261
set services ipsec-vpn rule demo-rule-r1-manual-sa term demo-term-manual-sa then manual
direction bidirectional authentication algorithm hmac-sha1-96
set services ipsec-vpn rule demo-rule-r1-manual-sa term demo-term-manual-sa then manual
direction bidirectional authentication key ascii-text demokeyipsecmanualsa
set services ipsec-vpn rule demo-rule-r1-manual-sa term demo-term-manual-sa then manual
direction bidirectional encryption algorithm des-cbc
set services ipsec-vpn rule demo-rule-r1-manual-sa term demo-term-manual-sa then manual
direction bidirectional encryption key ascii-text manualsa
set services ipsec-vpn rule demo-rule-r1-manual-sa match-direction input
set services service-set demo-ss-manual-sa next-hop-service inside-service-interface ms-1/2/0.1
set services service-set demo-ss-manual-sa next-hop-service outside-service-interface ms-1/2/0.2
set services service-set demo-ss-manual-sa ipsec-vpn-options local-gateway 10.1.15.1
set services service-set demo-ss-manual-sa ipsec-vpn-rules demo-rule-r1-manual-sa
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure OSPF connectivity and IPsec tunnel parameters on Router 2:

1. Configure interface properties. In this step, you configure two Ethernet interfaces (ge-1/0/0 and ge-1/0/1), a loopback interface, and a multiservices interface (ms-1/2/0).

   ```
   [edit interfaces]
   user@router2# set ge-1/0/0 unit 0 description "to R3 ge-1/0/0"
   user@router2# set ge-1/0/0 unit 0 family inet address 10.1.15.1/30
   user@router2# set ge-1/0/1 unit 0 description "to R1 ge-1/0/0"
   user@router2# set ge-1/0/1 unit 0 family inet address 10.1.12.2/30
   user@router2# set ms-1/2/0 unit 0 family inet
   user@router2# set ms-1/2/0 unit 1 family inet
   user@router2# set ms-1/2/0 unit 1 service-domain inside
   user@router2# set ms-1/2/0 unit 2 family inet
   ```

```
user@router2# set ms-1/2/0 unit 2 service-domain outside
user@router2# set lo0 unit 0 family inet address 10.0.0.2/32
```

2. Specify the OSPF area and associate the interfaces with the OSPF area.

```
[edit protocols]
user@router2# set ospf area 0.0.0.0 interface ge-1/0/1.0
user@router2# set ospf area 0.0.0.0 interface lo0.0
user@router2# set ospf area 0.0.0.0 interface ms-1/2/0.1
```

3. Configure the router ID.

```
[edit routing-options]
user@router2# set router-ID 10.0.0.2
```

4. Configure an IPsec rule. In this step, you configure an IPsec rule and specify manual SA parameters, such as the remote-gateway address, authentication and encryption properties, and so on.

```
[edit services ipsec-vpn]
user@router2# set rule demo-rule-r1-manual-sa term demo-term-manual-sa then remote-gateway
10.1.15.2
user@router2# set rule demo-rule-r1-manual-sa term demo-term-manual-sa then manual direction
bidirectional protocol esp
user@router2# set rule demo-rule-r1-manual-sa term demo-term-manual-sa then manual direction
bidirectional spi 261
user@router2# set rule demo-rule-r1-manual-sa term demo-term-manual-sa then manual direction
bidirectional authentication algorithm hmac-sha1-96
user@router2# set rule demo-rule-r1-manual-sa term demo-term-manual-sa then manual direction
bidirectional authentication key ascii-text demokeyipsecmanualsa
user@router2# set rule demo-rule-r1-manual-sa term demo-term-manual-sa then manual direction
bidirectional encryption algorithm des-cbc
user@router2# set rule demo-rule-r1-manual-sa term demo-term-manual-sa then manual direction
bidirectional encryption key ascii-text manualsa
user@router2# set rule demo-rule-r1-manual-sa match-direction input
```

5. Configure a next-hop style service set, specify the local-gateway address, and associate the IPsec VPN rule with the service set.

   ```
   [edit services]
   user@router2# set service-set demo-ss-manual-sa next-hop-service inside-service-interface
   ms-1/2/0.1
   user@router2# set service-set demo-ss-manual-sa next-hop-service outside-service-interface
   ms-1/2/0.2
   user@router2# set service-set demo-ss-manual-sa ipsec-vpn-options local-gateway 10.1.15.1
   user@router2# set service-set demo-ss-manual-sa ipsec-vpn-rules demo-rule-r1-manual-sa
   ```

6. Commit the configuration.

   ```
   [edit]
   user@router2# commit
   ```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols ospf`, `show routing-options`, and `show services` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration

```
user@router1# show interfaces
interfaces {
    ...
    ge-1/0/0 {
        unit 0 {
            description "to R3 ge-1/0/0";
            family inet {
                address 10.1.15.1/30;
            }
        }
    }
    ge-1/0/1 {
        unit 0 {
            description "to R1 ge-1/0/1";
            family inet {
                address 10.1.12.2/30;
            }
```

```
        }
    }
    ms-1/2/0 {
        unit 0 {
            family inet;
        }
        unit 1 {
            family inet;
            service-domain inside;
        }
        unit 2 {
            family inet;
            service-domain outside;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.2/32;
            }
        }
    }
    ...
}
```

```
user@router2# show protocols ospf
protocols {
    ospf {
        area 0.0.0.0 {
            interfaces ge-1/0/1.0;
            interface lo0;
            interface ms-1/2/0;
        }
    }
}
```

```
user@router2# show routing-options
routing-options {
    router-id 10.0.0.2;
}
```

```
user@router2# show services
services  {
    ipsec-vpn {
        rule demo-rule-r1-manual-sa {
            term demo-term-manual-sa {
                then {
                    remote-gateway 10.1.15.2;
                    manual {
                        direction bidirectional {
                            protocol esp;
                            spi 261;
                            authentication {
                                algorithm hmac-sha1-96;
                                key ascii-text "$ABC1223"; ## SECRET-DATA
                            }
                            encryption {
                                algorithm des-cbc;
                                key ascii-text "$ABC123"; ## SECRET-DATA
                            }
                        }
                    }
                }
            }
            match-direction input;
        }
    }
    service-set demo-ss-manual-sa {
        next-hop-service {
            inside-service-interface ms-1/2/0.1;
            outside-service-interface ms-1/2/0.2;
        }
        ipsec-vpn-options {
            local-gateway 10.1.15.1;
        }
        ipsec-vpn-rules demo-rule-r1-manual-sa;
    }
}
```

Configuring Router 3

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI, at the [edit] hierarchy level, of Router 3.

```
set interfaces ge-1/0/1 unit 0 description "to R4 ge-1/0/1"
set interfaces ge-1/0/0 unit 0 family inet address 10.1.56.1/30
set interfaces ge-1/0/0 unit 0 description "to R2 ge-1/0/0"
set interfaces ge-1/0/0 unit 0 family inet address 10.1.15.2/30
set interfaces ms-1/2/0 unit 0 family inet
set interfaces ms-1/2/0 unit 1 family inet
set interfaces ms-1/2/0 unit 1 service-domain inside
set interfaces ms-1/2/0 unit 2 family inet
set interfaces ms-1/2/0 unit 2 service-domain outside
set interfaces lo0 unit 0 family inet address 10.0.0.3/32
set protocols ospf area 0.0.0.0 interface ge-1/0/1.0
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface ms-1/2/0.1
set routing-options router-id 10.0.0.3
set services ipsec-vpn rule demo-rule-r1-manual-sa term demo-term-manual-sa then remote-gateway
10.1.15.1
set services ipsec-vpn rule demo-rule-r1-manual-sa term demo-term-manual-sa then manual
direction bidirectional protocol esp
set services ipsec-vpn rule demo-rule-r1-manual-sa term demo-term-manual-sa then manual
direction bidirectional spi 261
set services ipsec-vpn rule demo-rule-r1-manual-sa term demo-term-manual-sa then manual
direction bidirectional authentication algorithm hmac-sha1-96
set services ipsec-vpn rule demo-rule-r1-manual-sa term demo-term-manual-sa then manual
direction bidirectional authentication key ascii-text demokeyipsecmanualsa
set services ipsec-vpn rule demo-rule-r1-manual-sa term demo-term-manual-sa then manual
direction bidirectional encryption algorithm des-cbc
set services ipsec-vpn rule demo-rule-r1-manual-sa term demo-term-manual-sa then manual
direction bidirectional encryption key ascii-text manualsa
set services ipsec-vpn rule demo-rule-r1-manual-sa match-direction input
set services service-set demo-ss-manual-sa next-hop-service inside-service-interface ms-1/2/0.1
set services service-set demo-ss-manual-sa next-hop-service outside-service-interface ms-1/2/0.2
set services service-set demo-ss-manual-sa ipsec-vpn-options local-gateway 10.1.15.2
set services service-set demo-ss-manual-sa ipsec-vpn-rules demo-rule-r1-manual-sa
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure OSPF connectivity and IPsec tunnel parameters on Router 3:

1. Configure interface properties. In this step, you configure two Ethernet interfaces (ge-1/0/0 and ge-1/0/1), a loopback interface, and a multiservices interface (ms-1/2/0).

   ```
   [edit interfaces]
   user@router3# set ge-1/0/0 unit 0 description "to R4 ge-1/0/0"
   user@router3# set ge-1/0/0 unit 0 family inet address 10.1.56.1/30
   user@router3# set ge-1/0/1 unit 0 description "to R2 ge-1/0/1"
   user@router3# set ge-1/0/1 unit 0 family inet address 10.1.15.2/30
   user@router3# set ms-1/2/0 unit 0 family inet
   user@router3# set ms-1/2/0 unit 1 family inet
   user@router3# set ms-1/2/0 unit 1 service-domain inside
   user@router3# set ms-1/2/0 unit 2 family inet
   user@router3# set ms-1/2/0 unit 2 service-domain outside
   user@router3# set lo0 unit 0 family inet address 10.0.0.3/32
   ```

2. Specify the OSPF area and associate the interfaces with the OSPF area.

   ```
   [edit protocols]
   user@router3# set ospf area 0.0.0.0 interface ge-1/0/1.0
   user@router3# set ospf area 0.0.0.0 interface lo0.0
   user@router3# set ospf area 0.0.0.0 interface ms-1/2/0.1
   ```

3. Configure a router ID.

   ```
   [edit routing-options]
   user@router3# set router-id 10.0.0.3
   ```

4. Configure an IPsec rule. In this step, you configure an IPsec rule and specify manual SA parameters, such as the remote-gateway address, authentication and encryption properties, and so on.

   ```
   [edit services ipsec-vpn]
   user@router3# set rule demo-rule-r1-manual-sa term demo-term-manual-sa then remote-gateway
   ```

```
10.1.15.1
user@router3# set rule demo-rule-r1-manual-sa term demo-term-manual-sa then manual direction
bidirectional protocol esp
user@router3# set rule demo-rule-r1-manual-sa term demo-term-manual-sa then manual direction
bidirectional spi 261
user@router3# set rule demo-rule-r1-manual-sa term demo-term-manual-sa then manual direction
bidirectional authentication algorithm hmac-sha1-96
user@router3# set rule demo-rule-r1-manual-sa term demo-term-manual-sa then manual direction
bidirectional authentication key ascii-text demokeyipsecmanualsa
user@router3# set rule demo-rule-r1-manual-sa term demo-term-manual-sa then manual direction
bidirectional encryption algorithm des-cbc
user@router3# set rule demo-rule-r1-manual-sa term demo-term-manual-sa then manual direction
bidirectional encryption key ascii-text manualsa
user@router3# set rule demo-rule-r1-manual-sa match-direction input
```

5. Configure a next-hop style service set, specify the local-gateway address, and associate the IPsec VPN rule with the service set.

```
[edit services]
user@router3# set service-set demo-ss-manual-sa next-hop-service inside-service-interface
ms-1/2/0.1
user@router3# set service-set demo-ss-manual-sa next-hop-service outside-service-interface
ms-1/2/0.2
user@router3# set service-set demo-ss-manual-sa ipsec-vpn-options local-gateway 10.1.15.2
user@router3# set service-set demo-ss-manual-sa ipsec-vpn-rules demo-rule-r1-manual-sa
```

6. Commit the configuration.

```
[edit]
user@router3# commit
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols ospf`, `show routing-options`, and `show services` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration

```
user@router3# show interfaces
interfaces {
    ge-1/0/1 {
        unit 0 {
            description "to R4 ge-1/0/1";
            family inet {
                address 10.1.56.1/30;
            }
        }
    }
    ge-1/0/0 {
        unit 0 {
            description "to R2 ge-1/0/0";
            family inet {
                address 10.1.15.2/30;
            }
        }
    }
    ms-1/2/0 {
        unit 0 {
            family inet;
        }
        unit 1 {
            family inet;
            service-domain inside;
        }
        unit 2 {
            family inet;
            service-domain outside;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.3/32;
            }
        }
    }
}
```

```
user@router3# show protocols ospf
protocols {
    ospf {
        area 0.0.0.0 {
            interface ge-1/0/1.0;
            interface lo0.0;
            interface ms-1/2/0.1;
        }
    }
}
```

```
user@router3# show routing-options
routing-options {
    router-id 10.0.0.3;
}
```

```
user@router3# show services
services {
    ipsec-vpn {
        rule demo-rule-r1-manual-sa {
            term demo-term-manual-sa {
                then {
                    remote-gateway 10.1.15.1;
                    manual {
                        direction bidirectional {
                            protocol esp;
                            spi 261;
                            authentication {
                                algorithm hmac-sha1-96;
                                key ascii-text "$ABC123"; ## SECRET-DATA
                            }
                            encryption {
                                algorithm des-cbc;
                                key ascii-text "$ABC123"; ## SECRET-DATA
                            }
                        }
                    }
                }
            }
            match-direction input;
        }
    }
    service-set demo-ss-manual-sa {
        next-hop-service {
            inside-service-interface ms-1/2/0.1;
            outside-service-interface ms-1/2/0.2;
        }
        ipsec-vpn-options {
            local-gateway 10.1.15.2;
        }
        ipsec-vpn-rules demo-rule-r1-manual-sa;
    }
}
```

Configuring Router 4

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI, at the [edit] hierarchy level, of Router 4.

```
set interfaces ge-1/0/1 description "to R3 ge-1/0/1"
set interfaces ge-1/0/1 unit 0 family inet address 10.1.56.2/30
set interfaces lo0 unit 0 family inet address 10.0.0.4/32
set protocols ospf area 0.0.0.0 interface ge-1/0/1.0
set protocols ospf area 0.0.0.0 interface lo0.0
set routing-options router-id 10.0.0.4
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To set up OSPF connectivity with Router 3

1. Configure the interfaces. In this step, you configure an Ethernet interface (ge-1/0/1) and a loopback interface.

   ```
   user@router4# set interfaces ge-1/0/1 description "to R3 ge-1/0/1"
   user@router4# set interfaces ge-1/0/1 unit 0 family inet address 10.1.56.2/30
   user@router4# set interfaces lo0 unit 0 family inet address 10.0.0.4/32
   ```

2. Specify the OSPF area and associate the interfaces with the OSPF area.

```
user@router4# set protocols ospf area 0.0.0.0 interface ge-1/0/1.0
user@router4# set protocols ospf area 0.0.0.0 interface lo0.0
```

3. Configure the router ID.

```
[edit routing-options]
user@router4# set router-id 10.0.0.4
```

4. Commit the configuration.

```
[edit]
user@router4# commit
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols ospf`, and `show routing-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration

```
user@router4# show interfaces
interfaces {
    ge-1/0/1 {
        description "to R3 ge-1/0/1";
        unit 0 {
            family inet {
                address 10.1.56.2/30;
            }
        }
    }
    lo0{
        unit 0 {
            family inet {
                address 10.0.0.4/32;
            }
        }
    }
}
```

```
user@router4# show routing-options
routing-options {
    router-id 10.0.0.4;
}
```

```
user@router4# show protocols ospf
protocols {
    ospf {
        area 0.0.0.0 {
            interface lo0.0;
            interface ge-1/0/1.0;
        }
    }
}
```

Verification

IN THIS SECTION

- Verifying Traffic Flow Through the IPsec Tunnel | 671
- Verifying the Security Associations on Router 2 | 672
- Verifying the Security Associations on Router 3 | 673

To confirm that the manual SA configuration is working properly, perform the following tasks:

Verifying Traffic Flow Through the IPsec Tunnel

Purpose

Verify that the IPsec tunnel carries traffic between Router 1 and Router 4.

Action

Issue a `ping` command from Router 1 to `lo0` on Router 4.

```
user@router1> ping 10.0.0.4
PING 10.0.0.4 (10.0.0.4): 56 data bytes
64 bytes from 10.0.0.4: icmp_seq=0 ttl=254 time=1.375 ms
64 bytes from 10.0.0.4: icmp_seq=1 ttl=254 time=18.375 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=254 time=1.120 ms
^C
--- 10.0.0.4 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.120/6.957/18.375/8.075 ms
```

Meaning

The output shows that Router 1 is able to reach Router 4 over the IPsec tunnel.

Verifying the Security Associations on Router 2

Purpose

Verify that the security associations are active on Router 2 and that the traffic is flowing over the IPsec tunnel.

Action

- To verify that the security associations are active, Issue `show services ipsec-vpn ipsec security-associations detail` on Router 2.

  ```
  user@router2> show services ipsec-vpn ipsec security-associations detail
  Service set: demo-ss-manual-sa
  Rule: demo-rule-r1-manual-sa, Term: demo-term-manual-sa,
  Tunnel index: 1
  Local gateway: 10.1.15.1, Remote gateway: 10.1.15.2
  Local identity: ipv4_subnet(any:0,[0..7]=10.0.0.0/8)
  Remote identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Direction: inbound, SPI: 261, AUX-SPI: 0
  Mode: tunnel, Type: manual, State: Installed
  Protocol: ESP, Authentication: hmac-sha1-96, Encryption: des-cbc
  ```

```
Anti-replay service: Disabled
Direction: outbound, SPI: 261, AUX-SPI: 0
Mode: tunnel, Type: manual, State: Installed
Protocol: ESP, Authentication: hmac-sha1-96, Encryption: des-cbc
Anti-replay service: Disabled
```

- To verify that traffic is traveling over the bidirectional IPsec tunnel, issue `show services ipsec-vpn ipsec statistics` on Router 2.

```
user@router2> show services ipsec-vpn ipsec statistics
PIC: ms-1/2/0, Service set: demo-ss-manual-sa
sESP Statistics:
Encrypted bytes: 1616
Decrypted bytes: 1560
Encrypted packets: 20
Decrypted packets: 19
AH Statistics:
Input bytes: 0
Output bytes: 0
Input packets: 0
Output packets: 0
Errors:
AH authentication failures: 0, Replay errors: 0
ESP authentication failures: 0, ESP decryption failures: 0
Bad headers: 0, Bad trailers: 0
```

Meaning

The `show services ipsec-vpn ipsec security-associations detail` command output shows the SA properties that you configured.

The `show services ipsec-vpn ipsec statistics` command output shows the traffic flow over the IPsec tunnel.

Verifying the Security Associations on Router 3

Purpose

Verify the security associations and flow of traffic over the IPsec tunnel.

Action

- To verify that the security associations are active, Issue `show services ipsec-vpn ipsec security-associations detail` on Router 3.

```
user@router3> show services ipsec-vpn ipsec security-associations detail
Service set: demo-ss-manual-sa
Rule: demo-rule-r1-manual-sa, Term: demo-term-manual-sa,
Tunnel index: 1
Local gateway: 10.1.15.2, Remote gateway: 10.1.15.1
Local identity: ipv4_subnet(any:0,[0..7]=10.0.0.0/8)
Remote identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
Direction: inbound, SPI: 261, AUX-SPI: 0
Mode: tunnel, Type: manual, State: Installed
Protocol: ESP, Authentication: hmac-sha1-96, Encryption: des-cbc
Anti-replay service: Disabled
Direction: outbound, SPI: 261, AUX-SPI: 0
Mode: tunnel, Type: manual, State: Installed
Protocol: ESP, Authentication: hmac-sha1-96, Encryption: des-cbc
Anti-replay service: Disabled
```

- To verify that traffic is traveling over the bidirectional IPsec tunnel, issue `show services ipsec-vpn ipsec statistics` on Router 3.

```
user@router3> show services ipsec-vpn ipsec statistics
PIC: ms-1/2/0, Service set: demo-ss-manual-sa
ESP Statistics:
Encrypted bytes: 1560
Decrypted bytes: 1616
Encrypted packets: 19
Decrypted packets: 20
AH Statistics:
Input bytes: 0
Output bytes: 0
Input packets: 0
Output packets: 0
Errors:
AH authentication failures: 0, Replay errors: 0
ESP authentication failures: 0, ESP decryption failures: 0
Bad headers: 0, Bad trailers: 0
```

Meaning

The `show services ipsec-vpn ipsec security-associations detail` command output shows the SA properties that you configured.

The `show services ipsec-vpn ipsec statistics` command output shows the traffic flow over the IPsec tunnel.

SEE ALSO

Configuring Security Associations | 645

Example: Configuring IKE Dynamic SAs | 699

Dynamic Security Associations

IN THIS SECTION

- Configuring IKE Proposals | 675
- Configuring IKE Policies | 681
- Configuring IPsec Proposals | 688
- Configuring IPsec Policies | 694

Configuring IKE Proposals

IN THIS SECTION

- Configuring the Authentication Algorithm for an IKE Proposal | 676
- Configuring the Authentication Method for an IKE Proposal | 677
- Configuring the Diffie-Hellman Group for an IKE Proposal | 678
- Configuring the Encryption Algorithm for an IKE Proposal | 679
- Configuring the Lifetime for an IKE SA | 679
- Example: Configuring an IKE Proposal | 680

Dynamic security associations (SAs) require IKE configuration. With dynamic SAs, you configure IKE first, and then the SA. IKE creates the dynamic SAs and negotiates them for IPsec. The IKE configuration defines the algorithms and keys used to establish the secure IKE connection with the peer security gateway.

You can configure one or more IKE proposals. Each proposal is a list of IKE attributes to protect the IKE connection between the IKE host and its peer.

To configure an IKE proposal, include the `proposal` statement and specify a name at the `[edit services ipsec-vpn ike]` hierarchy level:

```
[edit services ipsec-vpn ike]
proposal proposal-name {
    authentication-algorithm (md5 | sha1 | sha-256);
    authentication-method  (ecdsa-signatures-256 | ecdsa-signatures-384 | pre-shared-keys | rsa-
signatures);
    dh-group (group1 | group2 | group5 |group14 | group 15 | group16 | group19 | group20 |
group24);
    encryption-algorithm algorithm;
    lifetime-seconds seconds;
}
```

NOTE: In Junos FIPS mode, ECDSA is not supported for the authentication method in Junos OS Release 17.3R1. Starting in Junos OS Release 17.4R1, ECDSA is supported in Junos FIPS mode.

This section includes the following topics:

Configuring the Authentication Algorithm for an IKE Proposal

To configure the authentication algorithm for an IKE proposal, include the `authentication-algorithm` statement at the `[edit services ipsec-vpn ike proposal *proposal-name*]` hierarchy level:

```
[edit services ipsec-vpn ike proposal proposal-name]
authentication-algorithm (md5 | sha1 | sha-256);
```

The authentication algorithm can be one of the following:

- `md5`—Produces a 128-bit digest.

- `sha1`—Produces a 160-bit digest.

- sha-256—Produces a 256-bit digest.

> **NOTE**: For reference information on Secure Hash Algorithms (SHAs), see Internet draft draft-eastlake-sha2-02.txt, *Secure Hash Algorithms (SHA and HMAC-SHA)* (expires July 2006).

Configuring the Authentication Method for an IKE Proposal

To configure the authentication method for an IKE proposal, include the authentication-method statement at the [edit services ipsec-vpn ike proposal *proposal-name*] hierarchy level:

```
[edit services ipsec-vpn ike proposal proposal-name]
authentication-method (ecdsa-signatures-256 | ecdsa-signatures-384 | pre-shared-keys | rsa-
signatures);
```

> **NOTE**: In IKEv1, the authentication method for SAs is negotiated with the remote peer based on the type of authentication method configured in the IKE proposal. In IKEv2, such a negotiation is not performed with the remote peer. Instead, each IKE peer uses the authentication method that is locally configured for them.
>
> For SAs in IKEv2, the authentication method is the default value as IKEv1 if an authentication method is not configured in the IKE proposal. If you are configuring an authentication method for IKEv2, you must have the same authentication method configured for all proposals referenced in the policy.

The authentication method can be one of the following:

> **NOTE**: In Junos FIPS mode, ECDSA is not supported for the authentication method in Junos OS Release 17.3R1. Starting in Junos OS Release 17.4R1, ECDSA is supported in Junos FIPS mode.

- ecdsa-signatures-256—Starting In Junos OS Release 17.3R1 for MS-MPCs and MS-MICs, Elliptic Curve Digital Signature Algorithm (ECDSA) for 256-bit moduli.
- ecdsa-signatures-384—Starting In Junos OS Release 17.3R1 for MS-MPCs and MS-MICs, Elliptic Curve Digital Signature Algorithm (ECDSA) for 384-bit moduli.
- pre-shared-keys—A key derived from an out-of-band mechanism; the key authenticates the exchanges.
- rsa-signatures—Public key algorithm (supports encryption and digital signatures).

Configuring the Diffie-Hellman Group for an IKE Proposal

Diffie-Hellman is a public-key cryptography scheme that allows two parties to establish a shared secret over an insecure communications channel. It is also used within IKE to establish session keys.

To configure the Diffie-Hellman group for an IKE proposal, include the `dh-group` statement at the `[edit services ipsec-vpn ike proposal proposal-name]` hierarchy level:

```
[edit services ipsec-vpn ike proposal proposal-name]
dh-group (group1 | group2 | group5 |group14 | group15 | group16 | group19 | group20 | group24);
```

The group can be one of the following:

- `group1`—Specifies that IKE uses the 768-bit Diffie-Hellman prime modulus group when performing the new Diffie-Hellman exchange.
- `group2`—Specifies that IKE uses the 1024-bit Diffie-Hellman prime modulus group when performing the new Diffie-Hellman exchange.
- `group5`—Specifies that IKE uses the 1536-bit Diffie-Hellman prime modulus group when performing the new Diffie-Hellman exchange.
- `group14`—Specifies that IKE uses the 2048-bit Diffie-Hellman prime modulus group when performing the new Diffie-Hellman exchange.
- `group19`—Specifies that IKE uses the 256-bit random Elliptic Curve Diffie-Hellman Group when performing the new Diffie-Hellman exchange.
- `group20`—-Specifies that IKE uses the 384-bit random Elliptic Curve Diffie-Hellman Group when performing the new Diffie-Hellman exchange.

Starting in Junos OS release 17.4R1, group15, group16, and group 24 can also be used:

- `group15`—Specifies that IKE use the 3072-bit Diffie-Hellman prime modulus group when performing the new Diffie-Hellman exchange.
- `group16`—Specifies that IKE use the 4096-bit Diffie-Hellman prime modulus group when performing the new Diffie-Hellman exchange.
- `group24`—Specifies that IKE use the 2048-bit Diffie-Hellman prime modulus group with 256-bit Prime Order Subgroup when performing the new Diffie-Hellman exchange.

Using a Diffie-Hellman group based on a greater number of bits results a more secure IKE tunnel than using a group based on fewer bits. However, this additional security might require additional processing time.

Configuring the Encryption Algorithm for an IKE Proposal

To configure the encryption algorithm for an IKE proposal, include the `encryption-algorithm` statement at the `[edit services ipsec-vpn ike proposal proposal-name]` hierarchy level:

```
[edit services ipsec-vpn ike proposal proposal-name]
encryption-algorithm algorithm;
```

The encryption algorithm can be one of the following:

- `3des-cbc`—Cipher block chaining encryption algorithm with a key size of 24 bytes; its key size is 192 bits long.
- `des-cbc`—Cipher block chaining encryption algorithm with a key size of 8 bytes; its key size is 56 bits long.
- `aes-128-cbc`—Advanced Encryption Standard (AES) 128-bit encryption algorithm.
- `aes-192-cbc`—Advanced Encryption Standard (AES) 192-bit encryption algorithm.
- `aes-256-cbc`—Advanced Encryption Standard (AES) 256-bit encryption algorithm.

NOTE: For a list of Data Encryption Standard (DES) encryption algorithm weak and semiweak keys, see RFC 2409, *The Internet Key Exchange (IKE)*. The AES encryption algorithms use a software implementation that has much lower throughput, so DES remains the recommended option.

For `3des-cbc`, the first 8 bytes should differ from the second 8 bytes, and the second 8 bytes should be the same as the third 8 bytes.

If you configure an authentication proposal but do not include the `encryption` statement, the result is NULL encryption. Certain applications expect this result. If you configure no specific authentication or encryption values, the Junos OS uses the default values of `sha1` for the authentication and `3des-cbc` for the encryption.

Configuring the Lifetime for an IKE SA

The `lifetime-seconds` statement sets the lifetime of an IKE SA. When the IKE SA expires, it is replaced by a new SA (and SPI) or the IPsec connection is terminated.

To configure the lifetime for an IKE SA, include the `lifetime-seconds` statement at the `[edit services ipsec-vpn ike proposal proposal-name]` hierarchy level:

```
[edit services ipsec-vpn ike proposal proposal-name]
lifetime-seconds seconds;
```

By default, the IKE SA lifetime is 3600 seconds. The range is from 180 through 86,400 seconds.

NOTE: In IKEv1, the lifetime for SAs is negotiated with the remote peer based on the type of lifetime configured in the IKE proposal. In IKEv2, such a negotiation is not performed with the remote peer. Instead, each IKE peer uses the lifetime that is locally configured for them.

For SAs in IKEv2, the lifetime is either the default value as IKEv1 (if another lifetime is not configured in the IKE proposal) or all IKEv2 proposals in the IKE policy must be configured with the same lifetime value.

NOTE: For IKE proposals, there is only one SA lifetime value, specified by the Junos OS. IPsec proposals use a different mechanism.

Example: Configuring an IKE Proposal

Configure an IKE proposal:

```
[edit services ipsec-vpn ike]
proposal ike-proposal {
    authentication-method pre-shared-keys;
    dh-group group1;
    authentication-algorithm sha1;
    encryption-algorithm 3des-cbc;
}
```

RELATED DOCUMENTATION

Configuring Security Associations | 645

Configuring IKE Policies

IN THIS SECTION

- Configuring the IKE Phase | 682
- Configuring the Mode for an IKE Policy | 683
- Configuring the Proposals in an IKE Policy | 683
- Configuring the Preshared Key for an IKE Policy | 683
- Configuring the Local Certificate for an IKE Policy | 684
- Configuring the Description for an IKE Policy | 685
- Configuring Local and Remote IDs for IKE Phase 1 Negotiation | 685
- Enabling Invalid SPI Recovery | 687
- Example: Configuring an IKE Policy | 687

An IKE policy defines a combination of security parameters (IKE proposals) to be used during IKE negotiation. It defines a peer address and the proposals needed for that connection. Depending on which authentication method is used, it defines the preshared key for the given peer or the local certificate. During the IKE negotiation, IKE looks for an IKE policy that is the same on both peers. The peer that initiates the negotiation sends all its policies to the remote peer, and the remote peer tries to find a match.

A match is made when both policies from the two peers have a proposal that contains the same configured attributes. If the lifetimes are not identical, the shorter lifetime between the two policies (from the host and peer) is used. The configured preshared key must also match its peer.

Starting with Junos OS Release 11.4, both IKEv1 and IKEv2 are supported by default on all M Series, MX Series, and T Series routers. You can configure the specific IKE phase to be supported for the negotiation. However, if only IKEv1 is supported, the Junos OS rejects IKEv2 negotiations. Similarly, if only IKEv2 is supported, the Junos OS rejects all IKEv1 negotiations.

The key management process (kmd) daemon determines which version of IKE is used in a negotiation. If kmd is the IKE initiator, it uses IKEv1 by default and retains the configured version for negotiations. If kmd is the IKE responder, it accepts connections from both IKEv1 and IKEv2.

You can create multiple, prioritized proposals at each peer to ensure that at least one proposal matches a remote peer’s proposal.

First, you configure one or more IKE proposals; then you associate these proposals with an IKE policy. You can also prioritize a list of proposals used by IKE in the `policy` statement by listing the proposals you want to use, from first to last.

To configure an IKE policy, include the `policy` statement and specify a policy name at the `[edit services ipsec-vpn ike]` hierarchy level:

```
[edit services ipsec-vpn ike]
policy policy-name {
    description description;
    local-certificate identifier;
    local-id (ipv4_addr ipv4-address | ipv6-addr ipv6-address | key-id identifier);
    version (1 | 2);
    mode (aggressive | main);
    pre-shared-key (ascii-text key | hexadecimal key);
    proposals [ proposal-names ];
    remote-id {
        any-remote-id;
        ipv4_addr [ values ];
        ipv6_addr [ values ];
        key_id [ values ];
    }
    respond-bad-spi max-responses;
}
```

This section includes the following topics:

Configuring the IKE Phase

Starting with Junos OS Release 11.4, both IKEv1 and IKEv2 are supported by default on all M Series, MX Series, and T Series routers. You can configure the specific IKE phase to be supported for the negotiation. However, if only IKEv1 is supported, the Junos OS rejects IKEv2 negotiations. Similarly, if only IKEv2 is supported, the Junos OS rejects all IKEv1 negotiations.

To configure the IKE phase used, include the `version` statement at the `[edit services ipsec-vpn ike policy policy-name]` hierarchy level:

```
[edit services ipsec-vpn ike policy policy-name]
version (1 | 2);
```

Configuring the Mode for an IKE Policy

IKE policy has two modes: aggressive and main. By default, main mode is enabled. Main mode uses six messages, in three exchanges, to establish the IKE SA. (These three steps are IKE SA negotiation, a Diffie-Hellman exchange, and authentication of the peer.) Main mode also allows a peer to hide its identity.

Aggressive mode also establishes an authenticated IKE SA and keys. However, aggressive mode uses half the number of messages, has less negotiation power, and does not provide identity protection. The peer can use the aggressive or main mode to start IKE negotiation; the remote peer accepts the mode sent by the peer.

NOTE: The mode configuration is required only if the `version` option is set to `1`.

To configure the mode for an IKE policy, include the `mode` statement and specify `aggressive` or `main` at the `[edit services ipsec-vpn ike policy` *`policy-name`*`]` hierarchy level:

```
[edit services ipsec-vpn ike policy policy-name]
mode (aggressive | main);
```

Configuring the Proposals in an IKE Policy

The IKE policy includes a list of one or more proposals associated with an IKE policy.

To configure the proposals in an IKE policy, include the `proposals` statement and specify one or more proposal names at the `[edit services ipsec-vpn ike policy` *`policy-name`*`]` hierarchy level:

```
[edit services ipsec-vpn ike policy policy-name]
proposals [ proposal-names ];
```

Configuring the Preshared Key for an IKE Policy

When you include the `authentication-method pre-shared-keys` statement at the `[edit services ipsec-vpn ike proposal` *`proposal-name`*`]` hierarchy level, IKE policy preshared keys authenticate peers. You must manually configure a preshared key, which must match that of its peer. The preshared key can be an ASCII text (alphanumeric) key or a hexadecimal key.

To configure the preshared key in an IKE policy, include the `pre-shared-key` statement and a key at the `[edit services ipsec-vpn ike policy policy-name]` hierarchy level:

```
[edit services ipsec-vpn ike policy policy-name]
pre-shared-key (ascii-text key | hexadecimal key);
```

The key can be one of the following:

- `ascii-text`—ASCII text key. With the `des-cbc` option, the key contains 8 ASCII characters. With the `3des-cbc` option, the key contains 24 ASCII characters.

- `hexadecimal`—Hexadecimal key. With the `des-cbc` option, the key contains 16 hexadecimal characters. With the `3des-cbc` option, the key contains 48 hexadecimal characters.

Configuring the Local Certificate for an IKE Policy

IN THIS SECTION

- Configuring a Certificate Revocation List | 685

When you include the `authentication-method rsa-signatures` statement at the `[edit services ipsec-vpn ike proposal proposal-name]` hierarchy level, public key infrastructure (PKI) digital certificates authenticate peers. You must identify a local certificate that is sent to the peer during the IKE authentication phase.

To configure the local certificate for an IKE policy, include the `local-certificate` statement at the `[edit services ipsec-vpn ike policy policy-name]` hierarchy level:

```
[edit services ipsec-vpn ike policy policy-name]
local-certificate identifier;
```

The `local-certificate` statement specifies the identifier used to obtain the end entity's certificate from the certification authority. Configuring it in an IKE policy allows you the flexibility of using a separate certificate with each remote peer if that is needed. You must also specify the identity of the certification authority by configuring the `ca-profile` statement at the `[edit security pki]` hierarchy level.

You can use the configured profiles to establish a set of trusted certification authorities for use with a particular service set. This enables you to configure separate service sets for individual clients to whom you are providing IP services; the distinct service sets provide logical separation of one set of IKE sessions from another, using different local gateway addresses, or *virtualization*. To configure the set of trusted certification authorities, include the `trusted-ca` statement at the `[edit services service-set service-set-name ipsec-vpn-options]` hierarchy level:

```
[edit services service-set service-set-name ipsec-vpn-options]
trusted-ca ca-profile;
```

See the following to configure a certificate revocation list:

Configuring a Certificate Revocation List

A certificate revocation list (CRL) contains a list of digital certificates that have been cancelled before their expiration date. When a participating peer uses a digital certificate, it checks the certificate signature and validity. It also acquires the most recently issued CRL and checks that the certificate serial number is not on that CRL.

> **NOTE**: By default, certificate revocation list verification is enabled. You can disable CRL verification by including the `disable` statement at the `[edit security pki ca-profile ca-profile-name revocation-check]` hierarchy level.
>
> By default, if the router either cannot access the Lightweight Directory Access Protocol (LDAP) URL or retrieve a valid certificate revocation list, certificate verification fails and the IPsec tunnel is not established. To override this behavior and permit the authentication of the IPsec peer when the CRL is not downloaded, include the `disable on-download-failure` statement at the `[edit security pki ca-profile ca-profile-name revocation-check crl]` hierarchy level.

To use the CA certificate revocation list, you include statements at the `[edit security pki ca-profile ca-profile-name revocation-check]` hierarchy level. For details, see the *Junos OS System Basics Configuration Guide*.

Configuring the Description for an IKE Policy

To specify an optional text description for an IKE policy, include the `description` statement at the `[edit services ipsec-vpn ike policy policy-name` hierarchy level:

```
[edit services ipsec-vpn ike policy policy-name]
description description;
```

Configuring Local and Remote IDs for IKE Phase 1 Negotiation

You can optionally specify local identifiers for use in IKE phase 1 negotiation. If the `local-id` statement is omitted, the local gateway address is used.

Starting with Junos OS Release 19.1R1, you can configure one of the local id type as distinguished name and you can configure one of the remote id type as distinguished name. The distinguished name field can be a container with container string values or wildcard with wildcard string values.

A distinguished name is a name used with digital certificates to uniquely identify a user. For example a distinguished name can be:

- CN=user
- DC=example
- DC=com

For the container string, the order of the fields and their values must exactly match the distinguished name in the peer's digital certificate. Example: `container ["C=US, ST=CA, L=Sunnyvale, O=Juniper, CN=local_neg, CN=test@juniper.net, OU=QA" "cn=admin, ou=eng, o=example, dc=net" ];`

For the wildcard string, the configured field and value must match the distinguished name in the peer's digital certificate but the order of the fields in the DN does not matter. Example: `wildcard [ "L=Sunnyvale, O=Juniper" "C=US, ST=CA" ];`

To specify one or more local IDs, include the `local-id` statement at the `[edit services ipsec-vpn ike policy` *`policy-name`*`]` hierarchy level:

```
[edit services ipsec-vpn ike policy policy-name]
local-id (distinguished-name container container-string-values |wildcard wildcard-string-values
fqdn fqdn-name ipv4_addr ipv4-address | ipv6-addr ipv6-address | key-id identifier);
```

You can also specify remote gateway identifiers for which the IKE policy is used. The remote gateway address in which this policy is defined is added by default.

To specify one or more remote IDs, include the `remote-id` statement at the `[edit services ipsec-vpn ike policy` *`policy-name`*`]` hierarchy level:

```
[edit services ipsec-vpn ike policy policy-name]
    remote-id {
    distinguished-name  container container-string-values |wildcard wildcard-string-values
    fqdn fqdn-name
    any-remote-id;
    ipv4_addr [ values ];
    ipv6_addr [ values ];
    key_id [ values ];
}
```

The `any-remote-id` option allows any remote address to connect. This option is supported only in dynamic endpoints configurations and cannot be configured along with specific values.

Enabling Invalid SPI Recovery

When peers in a security association (SA) become unsynchronized, packets with invalid security parameter index (SPI) values can be sent out, and the receiving peer drops these packets. For example, this could occur when one of the peers reboots. Starting in Junos OS Release 14.2, you can enable the device to recover when packets with invalid SPIs are received by resynchronizing the SAs.

To enable recovery from invalid SPI values, include the `respond-bad-spi` statement at the `[edit services ipsec-vpn ike policy]` *`policy-name`* hierarchy level:

```
[edit services ipsec-vpn ike policy policy-name]
respond-bad-spi  max-responses;
```

Example: Configuring an IKE Policy

Define two IKE policies: `policy 10.1.1.2` and `policy 10.1.1.1`. Each policy is associated with `proposal-1` and `proposal-2`. The following configuration uses only IKEv1 for negotiation.

```
[edit services ipsec-vpn]
ike {
    proposal proposal-1 {
        authentication-method pre-shared-keys;
        dh-group group1;
        authentication-algorithm sha1;
        encryption-algorithm 3des-cbc;
        lifetime-seconds 1000;
    }
    proposal proposal-2 {
        authentication-method pre-shared-keys;
        dh-group group2;
        authentication-algorithm md5;
        encryption-algorithm des-cbc;
        lifetime-seconds 10000;
    }
    proposal proposal-3 {
        authentication-method rsa-signatures;
        dh-group group2;
        authentication-algorithm md5;
        encryption-algorithm des-cbc;
        lifetime-seconds 10000;
    }
    policy 10.1.1.2 {
        mode main;
        proposals [ proposal-1 proposal-2 ];
        pre-shared-key ascii-text example-pre-shared-key;
    }
    policy 10.1.1.1 {
        local-certificate certificate-file-name;
        local-key-pair private-public-key-file;
        mode aggressive;
        proposals [ proposal-2 proposal-3 ]
        pre-shared-key hexadecimal 0102030abbcd;
    }
}
```

NOTE: Updates to the current IKE proposal and policy configuration are not applied to the current IKE SA; updates are applied to new IKE SAs.

If you want the new updates to take immediate effect, you must clear the existing IKE security associations so that they will be reestablished with the changed configuration. For information about how to clear the current IKE security association, see *clear services ipsec-vpn ike security-associations*.

RELATED DOCUMENTATION

Configuring Dynamic Endpoints for IPsec Tunnels | 817

Configuring Security Associations | 645

Configuring IPsec Proposals

IN THIS SECTION

- Configuring the Authentication Algorithm for an IPsec Proposal | 689
- Configuring the Description for an IPsec Proposal | 691
- Configuring the Encryption Algorithm for an IPsec Proposal | 691

- Configuring the Lifetime for an IPsec SA | 692
- Configuring the Protocol for a Dynamic SA | 693

An IPsec proposal lists protocols and algorithms (security services) to be negotiated with the remote IPsec peer.

To configure an IPsec proposal, include the `proposal` statement and specify an IPsec proposal name at the `[edit services ipsec-vpn ipsec]` hierarchy level:

```
[edit services ipsec-vpn ipsec]
proposal proposal-name {
    authentication-algorithm (hmac-md5-96 | hmac-sha1-96);
    description description;
    encryption-algorithm algorithm;
    lifetime-seconds seconds;
    protocol (ah | esp | bundle);
}
```

This section discusses the following topics:

Configuring the Authentication Algorithm for an IPsec Proposal

To configure the authentication algorithm for an IPsec proposal, include the `authentication-algorithm` statement at the `[edit services ipsec-vpn ipsec proposal proposal-name]` hierarchy level:

```
[edit services ipsec-vpn ipsec proposal proposal-name]
authentication-algorithm (hmac-md5-96 | hmac-sha1-96);
```

The authentication algorithm can be one of the following:

- `hmac-md5-96`—Hash algorithm that authenticates packet data. It produces a 128-bit digest. Only 96 bits are used for authentication.
- `hmac-sha1-96`—Hash algorithm that authenticates packet data. It produces a 160-bit digest. Only 96 bits are used for authentication.
- `hmac-sha-256-128`—Hash algorithm that authenticates packet data. Produces a 256-bit authenticator value.

NOTE: Keep the following points in mind when you configure the authentication algorithm in an IPsec proposal:

- When both ends of an IPsec VPN tunnel contain the same IKE proposal but different IPsec proposals, an error occurs and the tunnel is not established in this scenario. For example, if one end of the tunnel contains router 1 configured with the authentication algorithm as hmac-sha- 256-128 and the other end of the tunnel contains router 2 configured with the authentication algorithm as hmac-md5-96, the VPN tunnel is not established.

- When both ends of an IPsec VPN tunnel contain the same IKE proposal but different IPsec proposals, and when one end of the tunnel contains two IPsec proposals to check whether a less secure algorithm is selected or not, an error occurs and the tunnel is not established. For example, if you configure two authentication algorithms for an IPsec proposal as hmac-sha-256-128 and hmac-md5-96 on one end of the tunnel, router 1, and if you configure the algorithm for an IPsec proposal as hmac-md5-96 on the other end of the tunnel, router 2, the tunnel is not established and the number of proposals mismatch.

- When you configure two IPsec proposals at both ends of a tunnel, such as the `authentication-algorithm hmac-sha-256-128` and `authentication- algorithm hmac-md5-96` statements at the `[edit services ipsec-vpn ipsec proposal` *`proposal-name`*`]` hierarchy level on one of the tunnel, router 1 (with the algorithms in two successive statements to specify the order), and the `authentication-algorithm hmac-md5-96` and `authentication- algorithm hmac-sha-256-128` statements at the `[edit services ipsec-vpn ipsec proposal` *`proposal-name`*`]` hierarchy level on one of the tunnel, router 2 (with the algorithms in two successive statements to specify the order, which is the reverse order of router 1), the tunnel is established in this combination as expected because the number of proposals is the same on both ends and they contain the same set of algorithms. However, the authentication algorithm selected is hmac-md5-96 and not the stronger algorithm of hmac-sha-256-128. This method of selection of the algorithm occurs because the first matching proposal is selected. Also, for a default proposal, regardless of whether the router supports the Advanced Encryption Standard (AES) encryption algorithm, the 3des-cbc algorithm is chosen and not the aes-cfb algorithm, which is because of the first algorithm in the default proposal being selected. In the sample scenario described here, on router 2, if you reverse the order of the algorithm configuration in the proposal so that it is the same order as the one specified on router 1, hmac-sha-256-128 is selected as the authentication method.

- You must be aware of the order of proposals in an IPsec policy at the time of configuration if you want the matching of proposals to happen in a certain order of preference, such as the strongest algorithm to be considered first when a match is made when both policies from the two peers have a proposal.

Configuring the Description for an IPsec Proposal

To specify an optional text description for an IPsec proposal, include the `description` statement at the `[edit services ipsec-vpn ipsec proposal proposal-name]` hierarchy level:

```
[edit services ipsec-vpn ipsec proposal proposal-name]
description description;
```

Configuring the Encryption Algorithm for an IPsec Proposal

To configure encryption algorithm for an IPsec proposal, include the `encryption-algorithm` statement at the `[edit services ipsec-vpn ipsec proposal proposal-name]` hierarchy level:

```
[edit services ipsec-vpn ipsec proposal proposal-name]
encryption-algorithm algorithm;
```

The encryption algorithm can be one of the following:

- `3des-cbc`—Encryption algorithm that has a block size of 24 bytes; its key size is 192 bits long.
- `aes-128-cbc`—Advanced Encryption Standard (AES) 128-bit encryption algorithm.
- `aes-192-cbc`—Advanced Encryption Standard (AES) 192-bit encryption algorithm.
- `aes-256-cbc`—Advanced Encryption Standard (AES) 256-bit encryption algorithm.

> **NOTE**: In Junos FIPS mode, AES-GCM is not supported in Junos OS Release 17.3R1. Starting in Junos OS Release 17.4R1, AES-GCM is supported in Junos FIPS mode.

- `aes-128-gcm`—Starting In Junos OS Release 17.3R1 for MS-MPCs and MS-MICs, Advanced Encryption Standard in Galois/Counter Mode (AES-GCM) 128-bit encryption algorithm with a 16 octet integrity check value (ICV).
- `aes-192-gcm`—Starting In Junos OS Release 17.3R1 for MS-MPCs and MS-MICs, Advanced Encryption Standard in Galois/Counter Mode (AES-GCM) 192-bit encryption algorithm with a 16 octet integrity check value ICV.
- `aes-256-gcm`—Starting In Junos OS Release 17.3R1 for MS-MPCs and MS-MICs, Advanced Encryption Standard in Galois/Counter Mode (AES-GCM) 256-bit encryption algorithm with a 16 octet integrity check value ICV.

- des-cbc—Encryption algorithm that has a block size of 8 bytes; its key size is 48 bits long.

> **NOTE**: For a list of Data Encryption Standard (DES) encryption algorithm weak and semiweak keys, see RFC 2409, *The Internet Key Exchange (IKE)*. The AES encryption algorithms use a software implementation that has much lower throughput, so DES remains the recommended option.
>
> For 3des-cbc, the first 8 bytes should differ from the second 8 bytes, and the second 8 bytes should be the same as the third 8 bytes.
>
> If you do not configure specific authentication or encryption settings, Junos OS uses the default values of sha1 for the authentication and 3des-cbc for the encryption. For NULL encryption to be effective, you must always specify the Encapsulating Security Payload (ESP) protocol for the NULL encryption algorithm by including the protocol esp statement at the [edit services ipsec-vpn ipsec proposal *proposal-name*] hierarchy level, regardless of other system configurations.

Configuring the Lifetime for an IPsec SA

When a dynamic IPsec SA is created, two types of lifetimes are used: hard and soft. The hard lifetime specifies the lifetime of the SA. The soft lifetime, which is derived from the hard lifetime, informs the IPsec key management system that the SA is about to expire. This allows the key management system to negotiate a new SA before the hard lifetime expires.

> **NOTE**: In IKEv1, the lifetime for SAs is negotiated with the remote peer based on the type of lifetime configured in the IPsec proposal. In IKEv2, such a negotiation is not performed with the remote peer. Instead, each IKE peer uses the lifetime that is locally configured for them.
>
> For SAs in IKEv2, the lifetime is either the default value as IKEv1 (if another lifetime is not configured in the IPsec proposal) or all IKEv2 proposals in the IPsec policy must be configured with the same lifetime value.

To configure the hard lifetime value, include the lifetime-seconds statement and specify the number of seconds at the [edit services ipsec-vpn ipsec proposal *proposal-name*] hierarchy level:

```
[edit services ipsec-vpn ipsec proposal proposal-name]
lifetime-seconds seconds;
```

The default lifetime is 28,800 seconds. The range is from 180 through 86,400 seconds.

To calculate soft lifetime, lifetime-diff is calculated initially. Then based on whether the peer is the initiator or responder, soft lifetime is calculated.

The lifetime-diff calculation is performed as below:

- If (3*hard-lifetime)/10 is GREATER than 850 seconds, then lifetime-diff = 850 seconds + jitter between 0 to 850 seconds.

  NOTE: Jitter value increments from 0 to 850 on every IPsec SA install and will reset to 0.

- If (3*hard-lifetime)/10 is GREATER than 600 seconds and LESS than 850, then lifetime-diff = 600 seconds + random jitter between 0 to 45 seconds.
- If (3*hard-lifetime)/10 is GREATER than 90 seconds and LESS than 600, then lifetime-diff = 90 seconds + random jitter between 0 to 45 seconds.
- If (3*hard-lifetime)/10 is LESS than 90 seconds, then lifetime-diff = 90 seconds + random jitter between 0 to 10 seconds.

Based on the lifetime-diff, the soft lifetime is calculated as below:

- If lifetime-diff is GREATER than hard-lifetime, soft lifetime = (9*hard-lifetime)/10
- Initiator soft lifetime = hard-lifetime - lifetime-diff
- Responder soft lifetime = hard-lifetime - lifetime-diff + 45 seconds

NOTE: Initiator soft lifetime will be always less than responder soft lifetime. It is to ensure that initiator soft lifetime will expire first so that it can initiate rekey process.

For example, if hard lifetime is configured as 3600 seconds for IPSec SA's:

- Max soft lifetime of initiator is : 3600 - 850 (jitter equals 0) = 2750 seconds
- Min soft lifetime of initiator is : 3600 - 850 - 850 (jitter equals 850) = 1900 seconds
- Max soft lifetime of responder is : 3600 - 850 (jitter equals 0) + 45 = 2795 seconds
- Min soft lifetime of responder is : 3600 - 850 - 850 (jitter equals 850) + 45 = 1945 seconds

Configuring the Protocol for a Dynamic SA

The `protocol` statement sets the protocol for a dynamic SA. IPsec uses two protocols to protect IP traffic: ESP and AH. The ESP protocol can support authentication, encryption, or both. The AH protocol is used for strong authentication. AH also authenticates the IP packet. The `bundle` option uses AH authentication and ESP encryption; it does not use ESP authentication because AH provides stronger authentication of IP packets.

To configure the protocol for a dynamic SA, include the protocol statement and specify the ah, esp, or bundle option at the [edit services ipsec-vpn ipsec proposal *proposal-name*] hierarchy level:

```
[edit services ipsec-vpn ipsec proposal proposal-name]
protocol (ah | esp | bundle);
```

RELATED DOCUMENTATION

Configuring Security Associations | 645

Configuring IPsec Policies

IN THIS SECTION

- Configuring the Description for an IPsec Policy | 695
- Configuring Perfect Forward Secrecy | 695
- Configuring the Proposals in an IPsec Policy | 696
- IPsec Policy for Dynamic Endpoints | 696
- Example: Configuring an IPsec Policy | 697

An IPsec policy defines a combination of security parameters (IPsec proposals) used during IPsec negotiation. It defines Perfect Forward Secrecy (PFS) and the proposals needed for the connection. During the IPsec negotiation, IPsec looks for a proposal that is the same on both peers. The peer that initiates the negotiation sends all its policies to the remote peer, and the remote peer tries to find a match.

A match is made when both policies from the two peers have a proposal that contains the same configured attributes. If the lifetimes are not identical, the shorter lifetime between the two policies (from the host and peer) is used.

You can create multiple, prioritized IPsec proposals at each peer to ensure that at least one proposal matches a remote peer’s proposal.

First, you configure one or more IPsec proposals; then you associate these proposals with an IPsec policy. You can prioritize a list of proposals used by IPsec in the policy statement by listing the proposals you want to use, from first to last.

To configure an IPsec policy, include the `policy` statement, and specify the policy name and one or more proposals to associate with the policy, at the `[edit services ipsec-vpn ipsec]` hierarchy level:

```
[edit services ipsec-vpn ipsec]
policy policy-name {
    description description;
    perfect-forward-secrecy {
        keys (group1 | group2 | group5 | group14 |group15 |group16 | group24);
    }
    proposals [ proposal-names ];
}
```

This section includes the following topics related to configuring an IPsec policy:

Configuring the Description for an IPsec Policy

To specify an optional text description for an IPsec policy, include the `description` statement at the `[edit services ipsec-vpn ipsec policy policy-name]` hierarchy level:

```
[edit services ipsec-vpn ipsec policy policy-name]
description description;
```

Configuring Perfect Forward Secrecy

Perfect Forward Secrecy (PFS) provides additional security by means of a Diffie-Hellman shared secret value. With PFS, if one key is compromised, previous and subsequent keys are secure because they are not derived from previous keys. This statement is optional.

To configure PFS, include the `perfect-forward-secrecy` statement and specify a Diffie-Hellman group at the `[edit services ipsec-vpn ipsec policy policy-name]` hierarchy level:

```
[edit services ipsec-vpn ipsec policy policy-name]
perfect-forward-secrecy {
    keys (group1 | group2 | group5 | group14 | group15 |group16 | group24);
}
```

The key can be one of the following:

- `group1`—Specifies that IKE use the 768-bit Diffie-Hellman prime modulus group when performing the new Diffie-Hellman exchange.

- group2—Specifies that IKE use the 1024-bit Diffie-Hellman prime modulus group when performing the new Diffie-Hellman exchange.
- group5—Specifies that IKE use the 1536-bit Diffie-Hellman prime modulus group when performing the new Diffie-Hellman exchange.
- group14—Specifies that IKE use the 2048-bit Diffie-Hellman prime modulus group when performing the new Diffie-Hellman exchange.

Starting in Junos OS release 17.4R1, group15, group16, and group 24 can also be used for the key:

- group15—Specifies that IKE use the 3072-bit Diffie-Hellman prime modulus group when performing the new Diffie-Hellman exchange.
- group16—Specifies that IKE use the 4096-bit Diffie-Hellman prime modulus group when performing the new Diffie-Hellman exchange.
- group24—Specifies that IKE use the 2048-bit Diffie-Hellman prime modulus group with 256-bit Prime Order Subgroup when performing the new Diffie-Hellman exchange.

The higher numbered groups provide more security than the lowered numbered groups, but require more processing time.

Configuring the Proposals in an IPsec Policy

The IPsec policy includes a list of one or more proposals associated with an IPsec policy.

To configure the proposals in an IPsec policy, include the proposals statement and specify one or more proposal names at the [edit services ipsec-vpn ipsec policy *policy-name*] hierarchy level:

```
[edit services ipsec-vpn ipsec policy policy-name]
proposals [ proposal-names ];
```

IPsec Policy for Dynamic Endpoints

An IPsec policy for dynamic endpoints defines a combination of security parameters (IPsec proposals) used during IPsec negotiation between dynamic peer security gateways, in which the remote ends of tunnels do not have a statically assigned IP address. During the IPsec negotiation, the IPsec policy looks for an IPsec proposal that is the same on both peers. The peer that initiates the negotiation sends all its policies to the remote peer, and the remote peer tries to find a match. A match is made when the policies from the two peers have a proposal that contains the same configured attributes. If the lifetimes are not identical, the shorter lifetime between the two policies (from the host and peer) is used.

If no policy is set, any policy proposed by the dynamic peer is accepted.

Example: Configuring an IPsec Policy

Define an IPsec policy, `dynamic policy-1`, that is associated with two proposals (`dynamic-1` and `dynamic-2`):

```
[edit services ipsec-vpn ipsec]
proposal dynamic-1 {
    protocol esp;
    authentication-algorithm hmac-md5-96;
    encryption-algorithm 3des-cbc;
    lifetime-seconds 6000;
}
proposal dynamic-2 {
    protocol esp;
    authentication-algorithm hmac-sha1-96;
    encryption-algorithm 3des-cbc;
    lifetime-seconds 6000;
}
policy dynamic-policy-1 {
    perfect-forward-secrecy {
        keys group1;
    }
    proposals [ dynamic-1 dynamic-2 ];
}
```

NOTE: Updates to the current IPsec proposal and policy configuration are not applied to the current IPsec SA; updates are applied to new IPsec SAs.

If you want the new updates to take immediate effect, you must clear the existing IPsec security associations so that they will be reestablished with the changed configuration. For information about how to clear the current IPsec security association, see the *Junos OS System Basics and Services Command Reference*.

RELATED DOCUMENTATION

Configuring Security Associations | 645

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
17.4R1	Starting in Junos OS Release 17.4R1, ECDSA is supported in Junos FIPS mode.
17.4R1	Starting in Junos OS Release 17.4R1, ECDSA is supported in Junos FIPS mode.
17.4R1	Starting in Junos OS release 17.4R1, group15, group16, and group 24 can also be used
17.4R1	Starting in Junos OS Release 17.4R1, AES-GCM is supported in Junos FIPS mode.
17.4R1	Starting in Junos OS release 17.4R1, group15, group16, and group 24 can also be used for the key
17.3R1	Starting In Junos OS Release 17.3R1 for MS-MPCs and MS-MICs, Elliptic Curve Digital Signature Algorithm (ECDSA) for 256-bit moduli.
17.3R1	Starting In Junos OS Release 17.3R1 for MS-MPCs and MS-MICs, Elliptic Curve Digital Signature Algorithm (ECDSA) for 384-bit moduli.
17.3R1	Starting In Junos OS Release 17.3R1 for MS-MPCs and MS-MICs, Advanced Encryption Standard in Galois/Counter Mode (AES-GCM) 128-bit encryption algorithm with a 16 octet integrity check value (ICV).
17.3R1	Starting In Junos OS Release 17.3R1 for MS-MPCs and MS-MICs, Advanced Encryption Standard in Galois/Counter Mode (AES-GCM) 192-bit encryption algorithm with a 16 octet integrity check value ICV.
17.3R1	Starting In Junos OS Release 17.3R1 for MS-MPCs and MS-MICs, Advanced Encryption Standard in Galois/Counter Mode (AES-GCM) 256-bit encryption algorithm with a 16 octet integrity check value ICV.
14.2	Starting in Junos OS Release 14.2, you can enable the device to recover when packets with invalid SPIs are received by resynchronizing the SAs.

IPsec Rules and Rulesets

IN THIS SECTION

- Example: Configuring IKE Dynamic SAs | 699
- Configuring IPsec Rules | 723
- Configuring IPsec Rule Sets | 732

Example: Configuring IKE Dynamic SAs

IN THIS SECTION

- Requirements | 699
- Overview and Topology | 700
- Configuration | 701
- Verification | 717

This example shows how to configure IKE dynamic SAs and contains the following sections.

Requirements

This example uses the following hardware and software components:

- Four M Series, MX Series, or T Series routers with multiservices interfaces installed in them.
- Junos OS Release 9.4 or later.

No special configuration beyond device initiation is required before you can configure this feature.

Overview and Topology

IN THIS SECTION

- Topology | 700

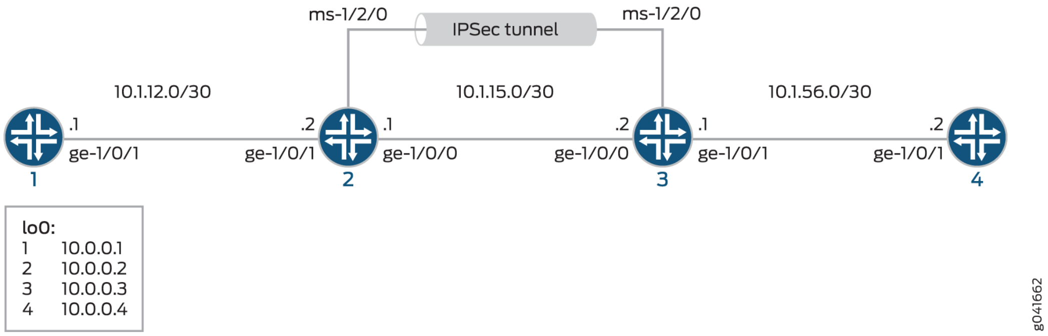
A security association (SA) is a simplex connection that enables two hosts to securely communicate with each other by means of IPsec.

Dynamic SAs are best suited for large-scale, geographically distributed networks where manual distribution, maintenance, and tracking of keys are difficult tasks. Dynamic SAs are configured with a set of proposals that are negotiated by the security gateways. The keys are generated as part of the negotiation and do not need to be specified in the configuration. A dynamic SA includes one or more proposals that allow you to prioritize a list of protocols and algorithms to be negotiated with the peer.

Topology

Figure 39 on page 700 shows an IPsec topology that contains a group of four routers. This configuration requires Routers 2 and 3 to establish an IPsec tunnel by using an IKE dynamic SA, enhanced authentication, and encryption. Routers 1 and 4 provide basic connectivity and are used to verify that the IPsec tunnel is operational.

Figure 39: IKE Dynamic SAs



NOTE: When you do not specify an IKE proposal, an IPsec proposal, and an IPsec policy on a MultiServices PIC, the Junos OS defaults to the highest level of encryption and authentication.

As a result, the default authentication protocol is ESP, the default authentication mode is HMAC-SHA1-96, and the default encryption mode is 3DES-CBC.

Configuration

IN THIS SECTION

- Configuring Router 1 | 701
- Configuring Router 2 | 704
- Configuring Router 3 | 710
- Configuring Router 4 | 715

To configure IKE dynamic SA, perform these tasks:

NOTE: The interface types shown in this example are for indicative purpose only. For example, you can use `so-` interfaces instead of `ge-` and `sp-` instead of `ms-`.

Configuring Router 1

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI, at the [edit] hierarchy level, of Router 1.

```
set interfaces ge-0/0/0 description "to R2 ge-0/0/0"
set interfaces ge-0/0/0 unit 0 family inet address 10.1.12.2/30
set interfaces lo0 unit 0 family inet address 10.0.0.1/32
set routing-options router-id 10.0.0.1
set protocols ospf area 0.0.0.0 interface ge-0/0/0
set protocols ospf area 0.0.0.0 interface lo0.0
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure Router 1 for OSPF connectivity with Router 2:

1. Configure an Ethernet interface and a loopback interface.

```
[edit interfaces]
user@router1# set ge-0/0/0 description "to R2 ge-0/0/0"
user@router1# set ge-0/0/0 unit 0 family inet address 10.1.12.2/30
user@router1# set lo0 unit 0 family inet address 10.0.0.1/32
```

2. Specify the OSPF area and associate the interfaces with the OSPF area.

```
[edit interfaces]
user@router1# set ospf area 0.0.0.0 interface ge-0/0/0.0
user@router1# set ospf area 0.0.0.0 interface lo0.0
```

3. Configure the router ID.

```
[edit routing-options]
user@router1# set router-id 10.0.0.1
```

4. Commit the configuration.

```
[edit]
user@router1# commit
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols ospf`, and `show routing-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration

```
user@router1# show interfaces
interfaces {
    ge-0/0/0 {
        description "To R2 ge-0/0/0";
        unit 0 {
            family inet {
                address 10.1.12.2/30;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.1/32;
            }
        }
    }
}
```

```
user@router1# show protocols ospf
protocols {
    ospf {
        area 0.0.0.0 {
            interface ge-0/0/0.0;
            interface lo0.0;
        }
    }
}
```

```
user@router1# show routing-options
routing-options  {
    router-id 10.0.0.1;
}
```

Configuring Router 2

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI, at the [edit] hierarchy level, of Router 2.

```
set interfaces ge-0/0/0 description "to R1 ge-0/0/0"
set interfaces ge-0/0/0 unit 0 family inet address 10.1.12.1/30
set interfaces ge-0/0/1 description "to R3 ge-0/0/1"
set interfaces ge-0/0/1 unit 0 family inet address 10.1.15.1/30
set interfaces ms-1/2/0 services-options syslog host local services info
set interfaces ms-1/2/0 unit 0 family inet
set interfaces ms-1/2/0 unit 1 family inet
set interfaces ms-1/2/0 unit 1 service-domain inside
set interfaces ms-1/2/0 unit 2 family inet
set interfaces ms-1/2/0 unit 2 service-domain outside
set interfaces lo0 unit 0 family inet address 10.0.0.2/32
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface ms-1/2/0.1
set routing-options router-id 10.0.0.2
set services ipsec-vpn rule rule-ike term term-ike then remote-gateway 10.1.15.2
set services ipsec-vpn rule rule-ike term term-ike then dynamic ike-policy ike-demo-policy
set services ipsec-vpn rule rule-ike term term-ike then dynamic ipsec-policy ipsec-demo-policy
set services ipsec-vpn rule match-direction input
set services ipsec-vpn ike proposal ike-demo-proposal authentication-method pre-shared-keys
set services ipsec-vpn ike proposal ike-demo-proposal dh-group group2
set services ipsec-vpn ike policy ike-demo-policy pre-shared proposals demo-proposal
set services ipsec-vpn ike policy ike-demo-policy pre-shared pre-shared-key ascii-text keyfordemo
set services ipsec-vpn ipsec proposal ipsec-demo-proposal protocol esp
set services ipsec-vpn ipsec proposal ipsec-demo-proposal authentication-algorithm hmac-sha1-96
set services ipsec-vpn ipsec proposal ipsec-demo-proposal encryption-algorithm 3des-cbc
set services ipsec-vpn ipsec policy ipsec-demo-policy perfect-forward-secrecy keys group2
set services ipsec-vpn ipsec proposals ipsec-demo-proposal
set services service-set demo-service-set next-hop-service inside-service-interface ms-1/2/0.1
set services service-set demo-service-set next-hop-service outside-service-interface ms-1/2/0.2
set services service-set demo-service-set ipsec-vpn-options local-gateway 10.1.15.1
set services service-set demo-service-set ipsec-vpn-rules rule-ike
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure OSPF connectivity and IPsec tunnel parameters on Router 2:

1. Configure interface properties. In this step, you configure two Ethernet interfaces (ge-1/0/0 and ge-1/0/1), a loopback interface, and a multiservices interface (ms-1/2/0).

```
[edit interfaces]
user@router2# set ge-0/0/0 description "to R1 ge-0/0/0"
user@router2# set ge-0/0/0 unit 0 family inet address 10.1.12.1/30
user@router2# set ge-0/0/1 description "to R3 ge-0/0/1"
user@router2# set ge-0/0/1 unit 0 family inet address 10.1.15.1/30
user@router2# set ms-1/2/0 services-options syslog host local services info
user@router2# set ms-1/2/0 unit 0 family inet
user@router2# set ms-1/2/0 unit 1 family inet
user@router2# set ms-1/2/0 unit 1 service-domain inside
user@router2# set ms-1/2/0 unit 2 family inet
user@router2# set ms-1/2/0 unit 2 service-domain outside
user@router2# set lo0 unit 0 family inet address 10.0.0.2/32
```

2. Specify the OSPF area and associate the interfaces with the OSPF area.

```
[edit protocols]
user@router2# set ospf area 0.0.0.0 interface ge-0/0/0.0
user@router2# set ospf area 0.0.0.0 interface lo0.0
user@router2# set ospf area 0.0.0.0 interface ms-1/2/0.1
```

3. Configure the router ID.

```
[edit routing-options]
user@router2# set router-ID 10.0.0.2
```

4. Configure an IPsec rule. In this step, you configure an IPsec rule, specify manual SA parameters, such as the remote gateway address, authentication and encryption properties, and so on.

NOTE: By default, Junos OS uses IKE policy version 1.0. Junos OS Release 11.4 and later also support IKE policy version 2.0 which you must configure at `[edit services ipsec-vpn ike policy` *`policy-name`* `pre-shared]`.

```
[edit services ipsec-vpn]
user@router2# set rule rule-ike term term-ike then remote-gateway 10.1.15.2
user@router2# set rule rule-ike term term-ike then dynamic ike-policy ike-demo-policy
user@router2# set rule rule-ike term term-ike then dynamic ipsec-policy ipsec-demo-policy
user@router2# set rule match-direction input
user@router2# set ike proposal ike-demo-proposal authentication-method pre-shared-keys
user@router2# set ike proposal ike-demo-proposal dh-group group2
user@router2# set ike policy ike-demo-policy pre-shared proposals demo-proposal
user@router2# set ike policy ike-demo-policy pre-shared pre-shared-key ascii-text keyfordemo
user@router2# set ipsec proposal ipsec-demo-proposal protocol esp
user@router2# set ipsec proposal ipsec-demo-proposal authentication-algorithm hmac-sha1-96
user@router2# set ipsec proposal ipsec-demo-proposal encryption-algorithm 3des-cbc
user@router2# set ipsec policy ipsec-demo-policy perfect-forward-secrecy keys group2
user@router2# set ipsec proposals ipsec-demo-proposal
```

5. Configure a next-hop style service set, specify the local-gateway address, and associate the IPsec VPN rule with the service set.

```
[edit services]
user@router2# set service-set demo-service-set next-hop-service inside-service-interface
ms-1/2/0.1
user@router2# set service-set demo-service-set next-hop-service outside-service-interface
ms-1/2/0.2
user@router2# set service-set demo-service-set ipsec-vpn-options local-gateway 10.1.15.1
user@router2# set service-set demo-service-set ipsec-vpn-rules rule-ike
```

6. Commit the configuration.

```
[edit]
user@router2# commit
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols ospf`, `show routing-options`, and `show services` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration

```
user@router1# show interfaces
interfaces {
    ge-0/0/0 {
        description "To R1 ge-0/0/0";
        unit 0 {
            family inet {
                address 10.1.12.1/30;
            }
        }
    }

    ge-0/0/1 {
        description "To R3 ge-0/0/1";
        unit 0 {
            family inet {
                address 10.1.15.1/30;
            }
        }
    }

    ms-1/2/0 {
        services-options {
            syslog {
                host local {
                    services info;
                }
            }
        }
        unit 0 {
            family inet;
        }
        unit 1 {
            family inet;
            service-domain inside;
        }
        unit 2 {
            family inet;
            service-domain outside;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.2/32;
            }
        }
    }
}
```

```
user@router2# show protocols ospf
protocols {
    ospf {
        area 0.0.0.0 {
            interface ge-0/0/0.0;
            interface lo0.0;
            interface ms-1/2/0.1;
        }
    }
}
```

```
user@router2# show routing-options
routing-options {
    router-id 10.0.0.2;
}
```

```
user@router2# show services
services {
    ipsec-vpn {
        rule rule-ike {
            term term-ike {
                then {
                    remote-gateway 10.1.15.2;
                    dynamic {
                        ike-policy ike-demo-policy;
                        ipsec-policy ipsec-demo-policy;
                    }
                }
            }
            match-direction input;
        }
        ike {
            proposal ike-demo-proposal {
                authentication-method pre-shared-keys;
                dh-group group2;
            }
            policy ike-demo-policy {
                proposals demo-proposal;
                pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
            }
        }
        ipsec {
        proposal ipsec-demo-proposal {
            protocol esp;
            authentication-algorithm hmac-sha1-96;
            encryption-algorithm 3des-cbc;
        }
        policy ipsec-demo-policy {
            perfect-forward-secrecy {
                keys group2;
            }
            proposals ipsec-demo-proposal;
        }
    }
}
service-set demo-service-set {
    next-hop-service {
        inside-service-interface ms-1/2/0.1;
        outside-service-interface ms-1/2/0.2;
    }
    ipsec-vpn-options {
        local-gateway 10.1.15.1;
    }
    ipsec-vpn-rules rule-ike;
}
service-set demo-service-set {
    next-hop-service {
        inside-service-interface ms-1/2/0.1;
        outside-service-interface ms-1/2/0.2;
    }
    ipsec-vpn-options {
        local-gateway 10.1.15.2;
    }
    ipsec-vpn-rules rule-ike;
}
```

Configuring Router 3

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI, at the [edit] hierarchy level, of Router 3.

```
set interfaces ge-0/0/0 description "to R4 ge-0/0/0"
set interfaces ge-0/0/0 unit 0 family inet address 10.1.56.1/30
set interfaces ge-0/0/1 description "to R2 ge-0/0/1"
set interfaces ge-0/0/1 unit 0 family inet address 10.1.15.2/30
set interfaces ms-1/2/0 services-options syslog host local services info
set interfaces ms-1/2/0 unit 0 family inet
set interfaces ms-1/2/0 unit 1 family inet
set interfaces ms-1/2/0 unit 1 service-domain inside
set interfaces ms-1/2/0 unit 2 family inet
set interfaces ms-1/2/0 unit 2 service-domain outside
set interfaces lo0 unit 0 family inet address 10.0.0.3/32
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface ms-1/2/0.1
set routing-options router-id 10.0.0.3
set services ipsec-vpn rule rule-ike term term-ike then remote-gateway 10.1.15.1
set services ipsec-vpn rule rule-ike term term-ike then dynamic ike-policy ike-demo-policy
set services ipsec-vpn rule rule-ike term term-ike then dynamic ipsec-policy ipsec-demo-policy
set services ipsec-vpn rule match-direction input
set services ipsec-vpn ike proposal ike-demo-proposal authentication-method pre-shared-keys
set services ipsec-vpn ike proposal ike-demo-proposal dh-group group2
set services ipsec-vpn ike policy ike-demo-policy pre-shared proposals demo-proposal
set services ipsec-vpn ike policy ike-demo-policy pre-shared pre-shared-key ascii-text keyfordemo
set services ipsec-vpn ipsec proposal ipsec-demo-proposal protocol esp
set services ipsec-vpn ipsec proposal ipsec-demo-proposal authentication-algorithm hmac-sha1-96
set services ipsec-vpn ipsec proposal ipsec-demo-proposal encryption-algorithm 3des-cbc
set services ipsec-vpn ipsec policy ipsec-demo-policy perfect-forward-secrecy keys group2
set services ipsec-vpn ipsec proposals ipsec-demo-proposal
set services service-set demo-service-set next-hop-service inside-service-interface ms-1/2/0.1
set services service-set demo-service-set next-hop-service outside-service-interface ms-1/2/0.2
set services service-set demo-service-set ipsec-vpn-options local-gateway 10.1.15.2
set services service-set demo-service-set ipsec-vpn-rules rule-ike
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure OSPF connectivity and IPsec tunnel parameters on Router 3:

1. Configure interface properties. In this step, you configure two Ethernet interfaces (ge-1/0/0 and ge-1/0/1), a loopback interface, and a multiservices interface (ms-1/2/0).

```
[edit interfaces]
user@router3# set ge-0/0/0 description "to R4 ge-0/0/0"
user@router3# set ge-0/0/0 unit 0 family inet address 10.1.56.1/30
user@router3# set ge-0/0/1 description "to R2 ge-0/0/1"
user@router3# set ge-0/0/1 unit 0 family inet address 10.1.15.2/30
user@router3# set ms-1/2/0 services-options syslog host local services info
user@router3# set ms-1/2/0 unit 0 family inet
user@router3# set ms-1/2/0 unit 1 family inet
user@router3# set ms-1/2/0 unit 1 service-domain inside
user@router3# set ms-1/2/0 unit 2 family inet
user@router3# set ms-1/2/0 unit 2 service-domain outside
user@router3# set lo0 unit 0 family inet address 10.0.0.3/32
```

2. Specify the OSPF area and associate the interfaces with the OSPF area.

```
[edit protocols]
user@router3# set ospf area 0.0.0.0 interface ge-0/0/0.0
user@router3# set ospf area 0.0.0.0 interface lo0.0
user@router3# set ospf area 0.0.0.0 interface ms-1/2/0.1
```

3. Configure a router ID.

```
[edit routing-options]
user@router3# set router-id 10.0.0.3
```

4. Configure an IPsec rule. In this step, you configure an IPsec rule and specify manual SA parameters, such as the remote gateway address, authentication and encryption properties, and so on.

```
[edit services ipsec-vpn]
user@router3# set rule rule-ike term term-ike then remote-gateway 10.1.15.1
user@router3# set rule rule-ike term term-ike then dynamic ike-policy ike-demo-policy
user@router3# set rule rule-ike term term-ike then dynamic ipsec-policy ipsec-demo-policy
user@router3# set rule match-direction input
user@router3#  set ike proposal ike-demo-proposal authentication-method pre-shared-keys
user@router3# set ike proposal ike-demo-proposal dh-group group2
user@router3# set ike policy ike-demo-policy pre-shared proposals demo-proposal
user@router3# set ike policy ike-demo-policy pre-shared pre-shared-key ascii-text keyfordemo
user@router3# set ipsec proposal ipsec-demo-proposal protocol esp
user@router3# set ipsec proposal ipsec-demo-proposal authentication-algorithm hmac-sha1-96
user@router3# set ipsec proposal ipsec-demo-proposal encryption-algorithm 3des-cbc
user@router3# set ipsec policy ipsec-demo-policy perfect-forward-secrecy keys group2
user@router3# set ipsec proposals ipsec-demo-proposal
```

5. Configure a next-hop style service set, specify the local-gateway address, and associate the IPsec VPN rule with the service set.

```
[edit services]
user@router3# set service-set demo-service-set next-hop-service inside-service-interface
ms-1/2/0.1
user@router3# set service-set demo-service-set next-hop-service outside-service-interface
ms-1/2/0.2
user@router3# set service-set demo-service-set ipsec-vpn-options local-gateway 10.1.15.2
user@router3# set service-set demo-service-set ipsec-vpn-rules rule-ike
```

6. Commit the configuration.

```
[edit]
user@router3# commit
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols ospf`, `show routing-options`, and `show services` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration

```
user@router3# show interfaces
interfaces {
    ge-0/0/0 {
        description "To R4 ge-0/0/0";
        unit 0 {
            family inet {
                address 10.1.56.1/30;
            }
        }
    }
    ge-0/0/1 {
        description "To R2 ge-0/0/1";
        unit 0 {
            family inet {
                address 10.1.15.2/30;
            }
        }
    }
    ms-1/2/0 {
        services-options {
            syslog {
                host local {
                    services info;
                }
            }
        }
        unit 0 {
            family inet {
            }
            unit 1 {
                family inet;
                service-domain inside;
            }
            unit 2 {
                family inet;
                service-domain outside;
            }
        }
        lo0 {
            unit 0 {
                family inet {
                    address 10.0.0.3/32;
                }
            }
        }
    }
}
```

```
user@router3# show protocols ospf
protocols {
    ospf {
        area 0.0.0.0 {
            interface ge-0/0/0.0;
            interface lo0.0;
            interface ms-1/2/0.1;
        }
    }
}
```

```
user@router3# show routing-options
routing-options {
    router-id 10.0.0.3;
}
```

```
user@router3# show services
services {
    ipsec-vpn {
        rule rule-ike {
            term term-ike {
                then {
                    remote-gateway 10.1.15.1;
                    dynamic {
                        ike-policy ike-demo-policy;
                        ipsec-policy ipsec-demo-policy;
                    }
                }
            }
            match-direction input;
        }
        ike {
            proposal ike-demo-proposal {
                authentication-method pre-shared-keys;
                dh-group group2;
            }
            policy ike-demo-policy {
                proposals demo-proposal;
                pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
            }
        }
        ipsec {
        proposal ipsec-demo-proposal {
            protocol esp;
            authentication-algorithm hmac-sha1-96;
            encryption-algorithm 3des-cbc;
        }
        policy ipsec-demo-policy {
            perfect-forward-secrecy {
                keys group2;
            }
            proposals ipsec-demo-proposal;
        }
    }
}
```

Configuring Router 4

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI, at the [edit] hierarchy level, of Router 4.

```
set interfaces ge-0/0/0 description "to R3 ge-0/0/0"
set interfaces ge-0/0/0 unit 0 family inet address 10.1.56.2/30
set interfaces lo0 unit 0 family inet address 10.0.0.4/32
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
set protocols ospf area 0.0.0.0 interface lo0.0
set routing-options router-id 10.0.0.4
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To set up OSPF connectivity with Router 4

1. Configure the interfaces. In this step, you configure an Ethernet interface (ge-1/0/1) and a loopback interface.

   ```
   user@router4# set interfaces ge-0/0/0 description "to R3 ge-0/0/0"
   user@router4# set interfaces ge-0/0/0 unit 0 family inet address 10.1.56.2/30
   user@router4# set interfaces lo0 unit 0 family inet address 10.0.0.4/32
   ```

2. Specify the OSPF area and associate the interfaces with the OSPF area.

   ```
   user@router4# set protocols ospf area 0.0.0.0 interface ge-0/0/0
   user@router4# set protocols ospf area 0.0.0.0 interface lo0.0
   ```

3. Configure the router ID.

   ```
   [edit routing-options]
   user@router4# set router-id 10.0.0.4
   ```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols ospf`, and `show routing-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration

```
user@router4# show interfaces
interfaces {
    ge-0/0/0 {
        description "To R3 ge-0/0/0";
```

```
        unit 0 {
            family inet {
                address 10.1.56.2/30;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.4/32;
            }
        }
    }
}
```

```
user@router4# show protocols ospf
protocols {
    ospf {
        area 0.0.0.0 {
            interface ge-0/0/0.0;
            interface lo0.0;
        }
    }
}
```

```
user@router4# show routing-options
routing-options {
    router-id 10.0.0.4;
}
```

Verification

IN THIS SECTION

- Verifying Your Work on Router 1 | **718**
- Verifying Your Work on Router 2 | **718**
- Verifying Your Work on Router 3 | **720**

- Verifying Your Work on Router 4 | 722

Verifying Your Work on Router 1

Purpose

Verify proper operation of Router 1.

Action

From operational mode, enter `ping 10.1.56.2` command to the ge-0/0/0 interface on Router 4 to send traffic across the IPsec tunnel

```
user@router1>ping 10.1.56.2
PING 10.1.56.2 (10.1.56.2): 56 data bytes
64 bytes from 10.1.56.2: icmp_seq=0 ttl=254 time=1.351 ms
64 bytes from 10.1.56.2: icmp_seq=1 ttl=254 time=1.187 ms
64 bytes from 10.1.56.2: icmp_seq=2 ttl=254 time=1.172 ms
64 bytes from 10.1.56.2: icmp_seq=3 ttl=254 time=1.154 ms
64 bytes from 10.1.56.2: icmp_seq=4 ttl=254 time=1.156 ms
^C
--- 10.1.56.2 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.154/1.204/1.351/0.074 ms
```

Meaning

The output shows that Router 1 is able to reach Router 4 over the IPsec tunnel.

Verifying Your Work on Router 2

Purpose

Verify that the IKE SA negotiation is successful.

Action

From operational mode, enter the `show services ipsec-vpn ike security-associations` command.

```
user@router2>show services ipsec-vpn ike security-associations
Remote Address State Initiator cookie Responder cookie Exchange type
10.1.15.2 Matured 03075bd3a0000003 4bff26a5c7000003 Main
```

To verify that the IPsec security association is active, issue the **show services ipsec-vpn ipsec security-associations detail** command. Notice that the SA contains the default settings inherent in the MultiServices PIC, such as ESP for the protocol and HMAC-SHA1-96 for the authentication algorithm.

From operational mode, enter the `show services ipsec-vpn ipsec security-associations detail` command.

```
user@router2> show services ipsec-vpn ipsec security-associations detail
Service set: demo-service-set
Rule: rule-ike, Term: term-ike, Tunnel index: 1
Local gateway: 10.1.15.1, Remote gateway: 10.1.15.2
Local identity: ipv4_subnet(any:0,[0..7]=10.1.12.0/24)
Remote identity: ipv4_subnet(any:0,[0..7]=10.1.56.0/24)
Direction: inbound, SPI: 2666326758, AUX-SPI: 0
Mode: tunnel, Type: dynamic, State: Installed
Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
Soft lifetime: Expires in 26863 seconds
Hard lifetime: Expires in 26998 seconds
Anti-replay service: Enabled, Replay window size: 64
Direction: outbound, SPI: 684772754, AUX-SPI: 0
Mode: tunnel, Type: dynamic, State: Installed
Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
Soft lifetime: Expires in 26863 seconds
Hard lifetime: Expires in 26998 seconds
Anti-replay service: Enabled, Replay window size: 64
```

To verify that traffic is traveling through the bidirectional IPsec tunnel, issue the **show services ipsec-vpn statistics** command:

From operational mode, enter the `show services ipsec-vpn statistics` command.

```
user@router2> show services ipsec-vpn ipsec statistics
PIC: ms-1/2/0, Service set: demo-service-set
ESP Statistics:
```

```
Encrypted bytes: 2248
Decrypted bytes: 2120
Encrypted packets: 27
Decrypted packets: 25
AH Statistics:
Input bytes: 0
Output bytes: 0
Input packets: 0
Output packets: 0
Errors:
AH authentication failures: 0, Replay errors: 0
ESP authentication failures: 0, ESP decryption failures: 0
Bad headers: 0, Bad trailers: 0
```

Meaning

The `show services ipsec-vpn ipsec security-associations detail` command output shows the SA properties that you configured.

The `show services ipsec-vpn ipsec statistics` command output shows the traffic flow over the IPsec tunnel.

Verifying Your Work on Router 3

Purpose

Verify that the IKE SA negotiation is successful on Router 3.

Action

From operational mode, enter the `show services ipsec-vpn ike security-associations` command. To be successful, the SA on Router 3 must contain the same settings you specified on Router 2.

```
user@router3>show services ipsec-vpn ike security-associations
Remote Address State Initiator cookie Responder cookie Exchange type
10.1.15.1 Matured 03075bd3a0000003 4bff26a5c7000003 Main
```

To verify that the IPsec SA is active, issue the **show services ipsec-vpn ipsec security-associations detail** command. To be successful, the SA on Router 3 must contain the same settings you specified on Router 2.

From operational mode, enter the `show services ipsec-vpn ipsec security-associations detail` command.

```
user@router3>show services ipsec-vpn ipsec security-associations detail
Service set: demo-service-set
Rule: rule-ike, Term: term-ike, Tunnel index: 1
Local gateway: 10.1.15.2, Remote gateway: 10.1.15.1
Local identity: ipv4_subnet(any:0,[0..7]=10.1.56.0/24)
Remote identity: ipv4_subnet(any:0,[0..7]=10.1.12.0/24)
Direction: inbound, SPI: 684772754, AUX-SPI: 0
Mode: tunnel, Type: dynamic, State: Installed
Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
Soft lifetime: Expires in 26598 seconds
Hard lifetime: Expires in 26688 seconds
Anti-replay service: Enabled, Replay window size: 64
Direction: outbound, SPI: 2666326758, AUX-SPI: 0
Mode: tunnel, Type: dynamic, State: Installed
Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
Soft lifetime: Expires in 26598 seconds
Hard lifetime: Expires in 26688 seconds
Anti-replay service: Enabled, Replay window size: 64
```

To verify that traffic is traveling through the bidirectional IPsec tunnel, issue the **show services ipsec-vpn statistics** command:

From operational mode, enter the `show services ipsec-vpn ike security-associations` command.

```
user@router3>show services ipsec-vpn ipsec statistics
PIC: ms-1/2/0, Service set: demo-service-set
ESP Statistics:
Encrypted bytes: 2120
Decrypted bytes: 2248
Encrypted packets: 25
Decrypted packets: 27
AH Statistics:
Input bytes: 0
Output bytes: 0
Input packets: 0
Output packets: 0
Errors:
AH authentication failures: 0, Replay errors: 0
```

```
ESP authentication failures: 0, ESP decryption failures: 0
Bad headers: 0, Bad trailers: 0
```

Meaning

The `show services ipsec-vpn ipsec security-associations detail` command output shows the SA properties that you configured.

The `show services ipsec-vpn ipsec statistics` command output shows the traffic flow over the IPsec tunnel.

Verifying Your Work on Router 4

Purpose

Verify that that the IKE SA negotiation is successful.

Action

From operational mode, enter `ping 10.1.12.2` command to the ge-0/0/0 interface on Router 1 to send traffic across the IPsec tunnel.

```
user@router4>ping 10.1.12.2
PING 10.1.12.2 (10.1.12.2): 56 data bytes
64 bytes from 10.1.12.2: icmp_seq=0 ttl=254 time=1.350 ms
64 bytes from 10.1.12.2: icmp_seq=1 ttl=254 time=1.161 ms
64 bytes from 10.1.12.2: icmp_seq=2 ttl=254 time=1.124 ms
64 bytes from 10.1.12.2: icmp_seq=3 ttl=254 time=1.142 ms
64 bytes from 10.1.12.2: icmp_seq=4 ttl=254 time=1.139 ms
64 bytes from 10.1.12.2: icmp_seq=5 ttl=254 time=1.116 ms
^C
--- 10.1.12.2 ping statistics ---
6 packets transmitted, 6 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.116/1.172/1.350/0.081 ms
```

To confirm that traffic travels through the IPsec tunnel, issue the `traceroute` command to the ge-0/0/0 interface on Router 1. Notice that the physical interface between Routers 2 and 3 is not referenced in the path; traffic enters the IPsec tunnel through the adaptive services IPsec inside interface on Router 3, passes through the loopback interface on Router 2, and ends at the ge-0/0/0 interface on Router 1.

From operational mode, enter the `traceroute 10.1.12.2`.

```
user@router4>traceroute 10.1.12.2
traceroute to 10.1.12.2 (10.1.12.2), 30 hops max, 40 byte packets
1 10.1.15.2 (10.1.15.2) 0.987 ms 0.630 ms 0.563 ms
2 10.0.0.2 (10.0.0.2) 1.194 ms 1.058 ms 1.033 ms
3 10.1.12.2 (10.1.12.2) 1.073 ms 0.949 ms 0.932 ms
```

Meaning

The `ping 10.1.12.2` output shows that Router 4 is able to reach Router 1 over the IPsec tunnel.

The `traceroute 10.1.12.2` output shows that traffic travels the IPsec tunnel.

SEE ALSO

- Understanding Junos VPN Site Secure | **623**
- Configuring Security Associations | **645**
- Configuring IKE Proposals | **675**
- Configuring IKE Policies | **681**
- Example: Configuring Manual SAs | **653**

Configuring IPsec Rules

IN THIS SECTION

- Configuring Match Direction for IPsec Rules | **725**
- Configuring Match Conditions in IPsec Rules | **726**
- Configuring Actions in IPsec Rules | **728**

To configure an IPsec rule, include the `rule` statement and specify a rule name at the `[edit services ipsec-vpn]` hierarchy level:

```
[edit services ipsec-vpn]
rule rule-name {
    match-direction (input | output);
    term term-name {
        from {
            destination-address address;
            ipsec-inside-interface interface-name;
            source-address address;
        }
        then {
            anti-replay-window-size bits;
            backup-remote-gateway address;
            clear-dont-fragment-bit;
            dynamic {
                ike-policy policy-name;
                ipsec-policy policy-name;
            }
            initiate-dead-peer-detection;
            dead-peer-detection {
                interval seconds;
                threshold number;
            }
            manual {
                direction (inbound | outbound | bidirectional) {
                    authentication {
                        algorithm (hmac-md5-96 | hmac-sha1-96);
                        key (ascii-text key | hexadecimal key);
                    }
                    auxiliary-spi spi-value;
                    encryption {
                        algorithm algorithm;
                        key (ascii-text key | hexadecimal key);
                    }
                    protocol (ah | bundle | esp);
                    spi spi-value;
                }
            }
            no-anti-replay;
            remote-gateway address;
            syslog;
            tunnel-mtu bytes;
        }
    }
}
```

Each IPsec rule consists of a set of terms, similar to a firewall filter.

A term consists of the following:

- `from` statement—Specifies the match conditions and applications that are included and excluded.
- `then` statement—Specifies the actions and action modifiers to be performed by the router software.

The following sections explain how to configure the components of IPsec rules:

Configuring Match Direction for IPsec Rules

Each rule must include a `match-direction` statement that specifies whether the match is applied on the input or output side of the interface. To configure where the match is applied, include the `match-direction (input | output)` statement at the `[edit services ipsec-vpn rule` *`rule-name`*`]` hierarchy level:

```
[edit services ipsec-vpn rule rule-name]
match-direction (input | output);
```

NOTE: ACX Series routers do not support `match-direction` as **output**.

The match direction is used with respect to the traffic flow through the AS or Multiservices PIC. When a packet is sent to the PIC, direction information is carried along with it.

With an interface service set, packet direction is determined by whether a packet is entering or leaving the interface on which the service set is applied.

With a next-hop service set, packet direction is determined by the interface used to route the packet to the AS or Multiservices PIC. If the inside interface is used to route the packet, the packet direction is input. If the outside interface is used to direct the packet to the PIC, the packet direction is output.

On the AS or Multiservices PIC, a flow lookup is performed. If no flow is found, rule processing is performed. All rules in the service set are considered. During rule processing, the packet direction is compared against rule directions. Only rules with direction information that matches the packet direction are considered.

Configuring Match Conditions in IPsec Rules

To configure the match conditions in an IPsec rule, include the `from` statement at the `[edit services ipsec-vpn rule rule-name term term-name]` hierarchy level:

```
[edit services ipsec-vpn rule rule-name term term-name]
from {
    destination-address address;
    ipsec-inside-interface interface-name;
    source-address address;
}
```

You can use either the source address or the destination address as a match condition, in the same way that you would configure a firewall filter; for more information, see the Junos OS Routing Protocols Library.

IPsec services support both IPv4 and IPv6 address formats. If you do not specifically configure either the source address or destination address, the default value `0.0.0.0/0` (IPv4 ANY) is used. To use IPv6 ANY (`0::0/128`) as either the source or destination address, you must configure it explicitly.

NOTE: IPsec services on ACX series support IPv4 address formats. If you do not specifically configure either the source address or destination address, the default value `0.0.0.0/0` (IPv4 ANY) is used.

For next-hop-style service sets only, the `ipsec-inside-interface` statement allows you to assign a logical interface to the tunnels established as a result of this match condition. The `inside-service-interface` statement that you can configure at the `[edit services service-set name next-hop-service]` hierarchy level allows you to specify `.1` and `.2` as inside and outside interfaces. However, you can configure multiple adaptive services logical interfaces with the `service-domain inside` statement and use one of them to configure the `ipsec-inside-interface` statement.

The Junos OS evaluates the criteria you configure in the `from` statement. If multiple link-type tunnels are configured within the same next-hop-style service set, the `ipsec-inside-interface` value enables the rule lookup module to distinguish a particular tunnel from other tunnels in case the source and destination addresses for all of them are `0.0.0.0/0` (ANY-ANY).

NOTE: When you configure the `ipsec-inside-interface` statement, interface-style service sets are not supported.

A special situation is provided by a term containing an “any-any” match condition (usually because the `from` statement is omitted). If there is an any-any match in a tunnel, a flow is not needed, because all flows within this tunnel use the same security association (SA) and packet selectors do not play a significant role. As a result, these tunnels will use packet-based IPsec. This strategy saves some flow resources on the PIC, which can be used for other tunnels that need a flow-based service.

The following configuration example shows an any-any tunnel configuration with no `from` statement in `term-1`. Missing selectors in the `from` clause result in a packet-based IPsec service.

```
services {
    ipsec-vpn {
        rule rule-1 {
            term term-1 {
                then {
                    remote-gateway 10.1.0.1;
                    dynamic {
                        ike-policy ike_policy;
                        ipsec-policy ipsec_policy;
                    }
                }
            }
            match-direction input;
        }
        .....
    }
```

Flowless IPsec service is provided to link-type tunnels with an any-any matching, as well as to dynamic tunnels with any-any matching in both dedicated and shared mode.

For link-type tunnels, a mixture of flowless and flow-based IPsec is supported within a service set. If a service set includes some terms with any-any matching and some terms with selectors in the `from` clause, packet-based service is provided for the any-any tunnels and flow-based service is provided for the other tunnels with selectors.

For non link-type tunnels, if a service set contains both any-any terms and selector-based terms, flow-based service is provided to all the tunnels.

Configuring Actions in IPsec Rules

IN THIS SECTION

- Enabling IPsec Packet Fragmentation | 729
- Configuring Destination Addresses for Dead Peer Detection | 729
- Configuring or Disabling IPsec Anti-Replay | 731
- Specifying the MTU for IPsec Tunnels | 732

To configure actions in an IPsec rule, include the `then` statement at the `[edit services ipsec-vpn rule` *`rule-name`* `term` *`term-name`*`]` hierarchy level:

```
[edit services ipsec-vpn rule rule-name term term-name]
then {
    anti-replay-window-size bits;
    backup-remote-gateway address;
    clear-dont-fragment-bit;
    dynamic {
        ike-policy policy-name;
        ipsec-policy policy-name;
    }
    initiate-dead-peer-detection;
    dead-peer-detection {
        interval seconds;
        threshold number;
    }
    manual {
        direction (inbound | outbound | bidirectional) {
            authentication {
                algorithm (hmac-md5-96 | hmac-sha1-96);
                key (ascii-text key | hexadecimal key);
            }
            auxiliary-spi spi-value;
            encryption {
                algorithm algorithm;
                key (ascii-text key | hexadecimal key);
            }
            protocol (ah | bundle | esp);
            spi spi-value;
        }
    }
    no-anti-replay;
    remote-gateway address;
    syslog;
    tunnel-mtu bytes;
}
```

The principal IPsec actions are to configure a dynamic or manual SA:

- You configure a dynamic SA by including the `dynamic` statement at the `[edit services ipsec-vpn rule rule-name term term-name then]` hierarchy level and referencing policies you have configured at the `[edit services ipsec-vpn ipsec]` and `[edit services ipsec-vpn ike]` hierarchy levels.

- You configure a manual SA by including the `manual` statement at the `[edit services ipsec-vpn rule rule-name term term-name then]` hierarchy level.

You can configure the following additional properties:

Enabling IPsec Packet Fragmentation

To enable fragmentation of IP version 4 (IPv4) packets in IPsec tunnels, include the `clear-dont-fragment-bit` statement at the `[edit services ipsec-vpn rule rule-name term term-name then]` hierarchy level:

```
[edit services ipsec-vpn rule rule-name term term-name then]
clear-dont-fragment-bit;
```

Setting the `clear-dont-fragment-bit` statement clears the Don’t Fragment (DF) bit in the packet header, regardless of the packet size. If the packet size exceeds the tunnel maximum transmission unit (MTU) value, the packet is fragmented before encapsulation. For IPsec tunnels, the default MTU value is 1500 regardless of the interface MTU setting.

Configuring Destination Addresses for Dead Peer Detection

To specify the remote address to which the IPsec traffic is directed, include the `remote-gateway` statement at the `[edit services ipsec-vpn rule rule-name term term-name then]` hierarchy level:

```
[edit services ipsec-vpn rule rule-name term term-name then]
remote-gateway address;
```

To specify a backup remote address, include the `backup-remote-gateway` statement at the `[edit services ipsec-vpn rule rule-name term term-name then]` hierarchy level:

```
[edit services ipsec-vpn rule rule-name term term-name then]
backup-remote-gateway address;
```

These two statements support both IPv4 and IPv6 address formats.

Configuring the `backup-remote-gateway` statement enables the dead peer detection (DPD) protocol, which monitors the tunnel state and remote peer availability. When the primary tunnel defined by the `remote-gateway` statement is active, the backup tunnel is in standby mode. If the DPD protocol determines that the primary remote gateway address is no longer reachable, a new tunnel is established to the backup address.

If there is no incoming traffic from a peer during a defined interval of 10 seconds, the router detects a tunnel as inactive. A global timer polls all tunnels every 10 seconds and the Adaptive Services (AS) or Multiservices Physical Interface Card (PIC) sends a message listing any inactive tunnels. If a tunnel becomes inactive, the router takes the following steps to fail over to the backup address:

1. The adaptive services message triggers the DPD protocol to send a hello message to the peer.

2. If no acknowledgment is received, two retries are sent at 2-second intervals, and then the tunnel is declared dead.

3. Failover takes place if the tunnel is declared dead or there is an IPsec Phase 1 negotiation timeout. The primary tunnel is put in standby mode and the backup becomes active.

4. If the negotiation to the backup tunnel times out, the router switches back to the primary tunnel. If both peers are down, it tries the failover six times. It then stops failing over and reverts to the original configuration, with the primary tunnel active and the backup in standby mode.

You can also enable triggering of DPD hello messages without configuring a backup remote gateway by including the `initiate-dead-peer-detection` statement at the `[edit services ipsec-vpn rule rule-name term term-name then]` hierarchy level:

```
[edit services ipsec-vpn rule rule-name term term-name then]
initiate-dead-peer-detection;
dead-peer-detection {
    interval seconds;
    threshold number;
}
```

In addition, for IKEv1 SAs you can set `interval` and `threshold` options under the `dead-peer-detection` statement when using the `initiate-dead-peer-detection` statement. Starting in Junos OS Release 17.2R1, the interval and threshold options are also applicable to IKEv2 SAs. In Junos OS Release 17.1 and earlier, the interval and threshold options are not applicable to IKEv2 SAs, which use the default values. The interval is the amount of time that the peer waits for traffic from its destination peer before sending a DPD request packet, and the threshold is the maximum number of unsuccessful DPD requests to be sent before the peer is considered unavailable.

The monitoring behavior is the same as described for the backup-remote-gateway statement. This configuration enables the router to initiate DPD hellos when a backup IPsec gateway does not exist, and clean up the IKE and IPsec SAs in case the IKE peer is not reachable.

If the DPD protocol determines that the primary remote gateway address is no longer reachable, a new tunnel is established to the backup address. However, when you configure initiate-dead-peer-detection without a backup remote gateway address and the DPD protocol determines that the primary remote gateway address is no longer reachable, the tunnel is declared dead and IKE and IPsec SAs are cleaned up.

For more information on the DPD protocol, see RFC 3706, *A Traffic-Based Method of Detecting Dead Internet Key Exchange (IKE) Peers*.

Configuring or Disabling IPsec Anti-Replay

To configure the size of the IPsec antireplay window, include the anti-replay-window-size statement at the [edit services ipsec-vpn rule *rule-name* term *term-name* then] hierarchy level:

```
[edit services ipsec-vpn rule rule-name term term-name then]
anti-replay-window-size bits;
```

anti-replay-window-size can take values in the range from 64 through 4096 bits. The default value is 64 bits for AS PICs and 128 bits for Multiservices PICs and DPCs. AS PICs can support a maximum replay window size of 1024 bits, whereas Multiservices PICs and DPCs can support a maximum replay window size of 4096 bits. When the software is committing an IPsec configuration , the key management process (kmd) is unable to differentiate between the service interface types. As a result, if the maximum antireplay window size exceeds 1024 for AS PICs, the commit succeeds and no error message is produced. However, the software internally sets the antireplay window size for AS PICs to 1024 bits even if the configured value of the anti-replay-window-size is larger.

To disable the IPsec antireplay feature, include the no-anti-replay statement at the [edit services ipsec-vpn rule *rule-name* term *term-name* then] hierarchy level:

```
[edit services ipsec-vpn rule rule-name term term-name then]
no-anti-replay;
```

By default, antireplay service is enabled. Occasionally this can cause interoperability issues with other vendors’ equipment.

Specifying the MTU for IPsec Tunnels

To configure a specific maximum transmission unit (MTU) value for IPsec tunnels, include the `tunnel-mtu` statement at the `[edit services ipsec-vpn rule rule-name term term-name then]` hierarchy level:

```
[edit services ipsec-vpn rule rule-name term term-name then]
tunnel-mtu bytes;
```

NOTE: The `tunnel-mtu` setting is the only place you need to configure an MTU value for IPsec tunnels. Inclusion of an `mtu` setting at the `[edit interfaces sp-fpc/pic/port unit logical-unit-number family inet]` hierarchy level is not supported.

RELATED DOCUMENTATION

Configuring Security Associations | 645

Configuring IPsec Rule Sets

The `rule-set` statement defines a collection of IPsec rules that determine what actions the router software performs on packets in the data stream. You define each rule by specifying a rule name and configuring terms. Then, you specify the order of the rules by including the `rule-set` statement at the `[edit services ipsec-vpn]` hierarchy level with a `rule` statement for each rule:

```
[edit services ipsec-vpn]
rule-set rule-set-name {
    rule rule-name;
}
```

The router software processes the rules in the order in which you specify them in the configuration. If a term in a rule matches the packet, the router performs the corresponding action and the rule processing stops. If no term in a rule matches the packet, processing continues to the next rule in the rule set. If none of the rules match the packet, the packet is dropped by default.

SEE ALSO

Configuring Security Associations | 645

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
17.2R1	Starting in Junos OS Release 17.2R1, the `interval` and `threshold` options are also applicable to IKEv2 SAs.

Service Sets for Static Endpoint IPsec Tunnels

IN THIS SECTION

- Service Sets | 733
- Configuring IPsec Service Sets | 734
- Requesting for and Installing a Digital Certificates on Your Router | 742
- Example: IKE Dynamic SA Configuration with Digital Certificates | 745
- Configuring Junos VPN Site Secure or IPSec VPN | 777
- Example: Configuring Junos VPN Site Secure on MS-MIC and MS-MPC | 778
- Example: Configuring Statically Assigned IPsec Tunnels over a VRF Instance | 793
- Multitask Example: Configuring IPsec Services | 801
- Disabling NAT-T on MX Series Routers for Handling NAT with IPsec-Protected Packets | 811
- Tracing Junos VPN Site Secure Operations | 812

Service Sets

The Adaptive Services PIC supports two types of service sets when you configure IPSec tunnels. Because they are used for different purposes, it is important to know the differences between these service set types.

- Next-hop service set—Supports multicast and multicast-style dynamic routing protocols (such as OSPF) over IPSec. Next-hop service sets allow you to use *inside* and *outside* logical interfaces on the Adaptive Services PIC to connect with multiple routing instances. They also allow the use of Network Address Translation (NAT) and stateful firewall capabilities. However, next-hop service sets do not monitor Routing Engine traffic by default and require configuration of multiple service sets to support traffic from multiple interfaces.

- Interface service set—Applied to a physical interface and similar to a stateless *firewall filter*. They are easy to configure, can support traffic from multiple interfaces, and can monitor Routing Engine traffic by default. However, they cannot support dynamic routing protocols or multicast traffic over the IPSec tunnel.

In general, we recommend that you use next-hop service sets because they support routing protocols and multicast over the IPSec tunnel, they are easier to understand, and the routing table makes forwarding decisions without administrative intervention.

SEE ALSO

Understanding Junos VPN Site Secure | **623**

Configuring Junos VPN Site Secure or IPSec VPN | **777**

Configuring IPsec Service Sets

IN THIS SECTION

- Configuring the Local Gateway Address for IPsec Service Sets | **735**
- Configuring IKE Access Profiles for IPsec Service Sets | **737**
- Configuring Certification Authorities for IPsec Service Sets | **738**
- Configuring or Disabling Antireplay Service | **738**
- Clearing the Do Not Fragment Bit | **739**
- Configuring Passive-Mode Tunneling | **740**
- Configuring the Tunnel MTU Value | **741**
- Configuring IPsec Multipath Forwarding with UDP Encapsulation | **741**

IPsec service sets require additional specifications that you configure at the `[edit services service-set` *`service-set-name`* `ipsec-vpn-options]` hierarchy level:

```
[edit services service-set service-set-name ipsec-vpn-options]
anti-replay-window-size bits;
clear-dont-fragment-bit;
copy-dont-fragment-bit
set-dont-fragment-bit
ike-access-profile profile-name;
```

```
local-gateway address <gw-interface interface-name.logical-unit-number>;
no-anti-replay;
no-certificate-chain-in-ike;
passive-mode-tunneling;
trusted-ca [ ca-profile-names ];
tunnel-mtu bytes;
```

Configuration of these statements is described in the following sections:

Configuring the Local Gateway Address for IPsec Service Sets

If you configure an IPsec service set, you must also configure a local IPv4 or IPv6 address by including the `local-gateway` statement:

- If the Internet Key Exchange (IKE) gateway IP address is in inet.0 (the default situation), you configure the following statement:

  ```
  local-gateway address;
  ```

- If the IKE gateway IP address is in a VPN routing and forwarding (VRF) instance, you configure the following statement:

  ```
  local-gateway address routing-instance instance-name;
  ```

You can configure all the link-type tunnels that share the same local gateway address in a single next-hop-style service set. You must specify a value for the `inside-service-interface` statement at the `[edit services service-set *service-set-name*]` hierarchy level that matches the `ipsec-inside-interface` value, which you configure at the `[edit services ipsec-vpn rule *rule-name* term *term-name* from]` hierarchy level. For more information about IPsec configuration, see "Configuring IPsec Rules" on page 723 .

NOTE: Starting in Junos OS Release 16.1, to configure link-type tunnels, (i.e., next-hop style), for HA purposes, you can configure AMS logical interfaces as the IPsec internal interfaces by using the `ipsec-inside-interface *interface-name*` statement at the `[edit services ipsec-vpn rule *rule-name* term *term-name* from]` hierarchy level.

Starting in Junos OS Release 17.1, AMS supports IPSec tunnel distribution.

IKE Addresses in VRF Instances

You can configure Internet Key Exchange (IKE) gateway IP addresses that are present in a VPN routing and forwarding (VRF) instance as long as the peer is reachable through the VRF instance.

For next-hop service sets, the key management process (kmd) places the IKE packets in the routing instance that contains the `outside-service-interface` value you specify, as in this example:

```
routing-instances vrf-nxthop {
    instance-type vrf;
    interface sp-1/1/0.2;
    ...
}
services service-set service-set-1 {
    next-hop-service {
        inside-service-interface sp-1/1/0.1;
        outside-service-interface sp-1/1/0.2;
    }
    ...
}
```

For interface service sets, the `service-interface` statement determines the VRF, as in this example:

```
routing-instances vrf-intf {
    instance-type vrf;
    interface sp-1/1/0.3;
    interface ge-1/2/0.1; # interface on which service set is applied
    ...
}
services service-set service-set-2 {
    interface-service {
        service-interface sp-1/1/0.3;
    }
    ...
}
```

Clearing SAs When Local Gateway Address or MS-MPC or MS-MIC Goes Down

Starting in Junos OS Release 17.2R1, tou can use the `gw-interface` statement to enable the cleanup of IKE triggers and IKE and IPsec SAs when an IPsec tunnel's local gateway IP address goes down, or the MS-MIC or MS-MPC being used in the tunnel's service set goes down.

```
local-gateway address <gw-interface interface-name.logical-unit-number>;
```

The *interface-name* and *logical-unit-number* must match the interface and logical unit on which the local gateway IP address is configured.

If the local gateway IP address for an IPsec tunnel's service set goes down or the MS-MIC or MS-MPC that is being used in the service set goes down, the service set no longer sends IKE triggers. In addition, when the local gateway IP address goes down, the IKE and IPsec SAs are cleared for next-hop service sets, and go to the Not Installed state for interface-style service sets. The SAs that have the Not Installed state are deleted when the local gateway IP address comes back up.

If the local gateway IP address that goes down for a next-hop service set is for the responder peer, then you need to clear the IKE and IPsec SAs on the initiator peer so that the IPsec tunnel comes back up once the local gateway IP address comes back up. You can either manually clear the IKE and IPsec SAs on the initiator peer (see *clear services ipsec-vpn ike security-associations* and *clear services ipsec-vpn ipsec security-associations*) or enable dead peer detection on the initiator peer (see "Configuring Stateful Firewall Rules" on page 550).

Configuring IKE Access Profiles for IPsec Service Sets

For dynamic endpoint tunneling only, you need to reference the IKE access profile configured at the `[edit access]` hierarchy level. To do this, include the `ike-access-profile` statement at the `[edit services service-set service-set-name ipsec-vpn-options]` hierarchy level:

```
[edit services service-set service-set-name ipsec-vpn-options]
ike-access-profile profile-name;
```

The `ike-access-profile` statement must reference the same name as the `profile` statement you configured for IKE access at the `[edit access]` hierarchy level. You can reference only one access profile in each service set. This profile is used to negotiate IKE and IPsec security associations with dynamic peers only.

NOTE: If you configure an IKE access profile in a service set, no other service set can share the same `local-gateway` address.

Also, you must configure a separate service set for each VRF. All interfaces referenced by the ipsec-inside-interface statement within a service set must belong to the same VRF.

Configuring Certification Authorities for IPsec Service Sets

You can specify one or more trusted certification authorities by including the trusted-ca statement:

```
trusted-ca [ ca-profile-names ];
```

When you configure public key infrastructure (PKI) digital certificates in the IPsec configuration, each service set can have its own set of trusted certification authorities. The names you specify for the trusted-ca statement must match profiles configured at the [edit security pki] hierarchy level; for more information, see the Junos OS Administration Library for Routing Devices. For more information about IPsec digital certificate configuration, see "Configuring IPsec Rules" on page 723 .

Starting in Junos OS Release 18.2R1, you can configure the MX Series router with MS-MPCs or MS-MICs to send only the end-entity certificate for certificate-based IKE authentication instead of the full certificate chain. This avoids IKE fragmentation. To configure this feature, include the no-certificate-chain-in-ike statement:

```
[edit services service-set service-set-name ipsec-vpn-options]
no-certificate-chain-in-ike;
```

Configuring or Disabling Antireplay Service

You can include the anti-replay-window-size statement at the [edit services service-set *service-set-name* ipsec-vpn-options] hierarchy level to specify the size of the antireplay window.

```
anti-replay-window-size bits;
```

This statement is useful for dynamic endpoint tunnels for which you cannot configure the anti-replay-window-size statement at the [edit services ipsec-vpn rule *rule-name* term *term-name* then] hierarchy level.

For static IPsec tunnels, this statement sets the antireplay window size for all the static tunnels within this service set. If a particular tunnel needs a specific value for antireplay window size, set the anti-replay-window-size statement at the [edit services ipsec-vpn rule *rule-name* term *term-name* then] hierarchy level. If antireplay check has to be disabled for a particular tunnel in this service set, set the no-anti-replay statement at the [edit services ipsec-vpn rule *rule-name* term *term-name* then] hierarchy level.

NOTE: The anti-replay-window-size and no-anti-replay settings at the [edit services ipsec-vpn rule *rule-name* term *term-name* then] hierarchy level override the settings specified at the [edit services service-set *service-set-name* ipsec-vpn-options] hierarchy level.

You can also include the no-anti-replay statement at the [edit services service-set *service-set-name* ipsec-vpn-options] hierarchy level to disable IPsec antireplay service. It occasionally causes interoperability issues for security associations.

```
no-anti-replay;
```

This statement is useful for dynamic endpoint tunnels for which you cannot configure the no-anti-reply statement at the [edit services ipsec-vpn rule *rule-name* term *term-name* then] hierarchy level.

For static IPsec tunnels, this statement disables the antireplay check for all the tunnels within this service set. If antireplay check has to be enabled for a particular tunnel, then set the anti-replay-window-size statement at the [edit services ipsec-vpn rule *rule-name* term *term-name* then] hierarchy level.

NOTE: Setting the anti-replay-window-size and no-anti-replay statements at the [edit services ipsec-vpn rule *rule-name* term *term-name* then] hierarchy level overrides the settings specified at the [edit services service-set *service-set-name* ipsec-vpn-options] hierarchy level.

Clearing the Do Not Fragment Bit

You can include the clear-dont-fragment-bit statement at the [edit services service-set *service-set-name* ipsec-vpn-options] hierarchy level to clear the do not fragment (DF) bit on all IP version 4 (IPv4) packets entering the IPsec tunnel. If the encapsulated packet size exceeds the tunnel maximum transmission unit (MTU), the packet is fragmented before encapsulation.

```
clear-dont-fragment-bit;
```

This statement is useful for dynamic endpoint tunnels for which you cannot configure the clear-dont-fragment-bit statement at the [edit services ipsec-vpn rule *rule-name* term *term-name* then] hierarchy level.

For static IPsec tunnels, setting this statement clears the DF bit on packets entering all the static tunnels within this service set. If you want to clear the DF bit on packets entering a specific tunnel, set the clear-dont-fragment-bit statement at the [edit services ipsec-vpn rule *rule-name* term *term-name* then] hierarchy level.

Starting in Junos OS Release 14.1, in packets that are transmitted through dynamic endpoint IPSec tunnels, you can enable the value set in the DF bit of the packet entering the tunnel to be copied only to the outer header of the IPsec packet and to not cause any modification to the DF bit in the inner header of the IPsec packet. If the packet size exceeds the tunnel maximum transmission unit (MTU) value, the packet is fragmented before encapsulation. For IPsec tunnels, the default MTU value is 1500 regardless of the interface MTU setting. To copy the DF bit value to only the outer header and not modify the inner header, use the `copy-dont-fragment-bit` statement at the `[edit services service-set` *`service-set-name`* `ipsec-vpn-options]` hierarchy level. You can also configure the DF bit to be set only in the outer IPv4 header of the IPsec packet and not be defined in the inner IPv4 header. To configure the DF bit in only the outer header of the IPsec packet and to leave the inner header unmodified, include the `set-dont-fragment-bit` statement at the `[edit services service-set` *`service-set-name`* `ipsec-vpn-options]` hierarchy level. These settings apply for dynamic endpoint tunnels and not for static tunnels, for which you need to include the `copy-dont-fragment-bit` and `set-dont-fragment-bit` statements at the `[edit services ipsec-vpn rule` *`rule-name`* `term` *`term-name`* `then]` hierarchy level to clear the DF bit in the IPv4 packets that enter the static tunnel. These functionalities are supported on MX Series routers with MS-MICs and MS-MPCs.

Configuring Passive-Mode Tunneling

You can include the `passive-mode-tunneling` statement at the `[edit services service-set` *`service-set-name`* `ipsec-vpn-options]` hierarchy level to enable the service set to tunnel malformed packets.

```
[edit services service-set service-set-name ipsec-vpn-options]
passive-mode-tunneling;
```

This functionality bypasses the active IP checks, such as version, TTL, protocol, options, address and other land attack checks, and tunnels the packets as is. If this statement is not configured, packets failing the IP checks are dropped in the PIC. In passive mode, the inner packet is not touched; an ICMP error is not generated if the packet size exceeds the tunnel MTU value.

The IPsec tunnel is not treated as a next hop and TTL is not decremented. Because an ICMP error is not generated if the packet size exceeds the tunnel MTU value, the packet is tunnelled even if it crosses the tunnel MTU threshold.

NOTE: This functionality is similar to that provided by the `no-ipsec-tunnel-in-traceroute` statement, described in "Tracing Junos VPN Site Secure Operations" on page 812 . Starting in Junos OS Release 14.2, passive mode tunneling is supported on MS-MICs and MS-MPCs.

NOTE: Starting in Junos OS Release 14.2, the `header-integrity-check` option that is supported on MS-MICs and MS-MPCs to verify the packet header for anomalies in IP, TCP, UDP, and ICMP information and flag such anomalies and errors has a functionality that is opposite to the functionality caused by passive mode tunneling. If you configure both the `header-integrity-check` statement and the `passive-mode tunneling` statement on MS-MICs and MS-MPCs, and attempt to commit such a configuration, an error is displayed during commit.

The passive mode tunneling functionality (by including the `passive-mode-tunnelin` statement at the `[edit services service-set` *service-set-name* `ipsec-vpn-options]` hierarchy level) is a superset of the capability to disable IPsec tunnel endpoint in the traceroute output (by including `no-ipsec-tunnel-in-traceroute` statement at the `[edit services ipsec-vpn]` hierarchy level). Passive mode tunneling also bypasses the active IP checks and tunnel MTU check in addition to not treating an IPsec tunnel as a next-hop as configured by the `no-ipsec-tunnel-in-traceroute` statement.

Configuring the Tunnel MTU Value

You can include the `tunnel-mtu` statement at the `[edit services service-set` *service-set-name* `ipsec-vpn-options]` hierarchy level to set the maximum transmission unit (MTU) value for IPsec tunnels.

```
tunnel-mtu bytes;
```

This statement is useful for dynamic endpoint tunnels for which you cannot configure the `tunnel-mtu` statement at the `[edit services ipsec-vpn rule` *rule-name* `term` *term-name* `then]` hierarchy level.

For static IPsec tunnels, this statement sets the tunnel MTU value for all the tunnels within this service set. If you need a specific value for a particular tunnel, then set the `tunnel-mtu` statement at the `[edit services ipsec-vpn rule` *rule-name* `term` *term-name* `then]` hierarchy level.

NOTE: The `tunnel-mtu` setting at the `[edit services ipsec-vpn rule` *rule-name* `term` *term-name* `then]` hierarchy level overrides the value specified at the `[edit services service-set` *service-set-name* `ipsec-vpn-options]` hierarchy level.

Configuring IPsec Multipath Forwarding with UDP Encapsulation

Starting in Junos OS Release 16.1, you can enable multipath forwarding of IPsec traffic by configuring UDP encapsulation in the service set, which adds a UDP header to the IPsec encapsulation of packets. This results in the forwarding of IPsec traffic over multiple paths, increasing the throughput of IPsec traffic. If you do not enable UDP encapsulation, all the IPsec traffic follows a single forwarding path.

When NAT-T is detected, only NAT-T UDP encapsulation occurs, not the UDP encapsulation for IPsec packets.

To enable UDP encapsulation:

1. Enable UDP encapsulation.

   ```
   [edit services service-set service-set-name ipsec-vpn-options]
   user@host set udp-encapsulation
   ```

2. (Optional) Specify the UDP destination port number.

   ```
   [edit services service-set service-set-name ipsec-vpn-options udp-encapsulation]
   user@host set udp-dest-port destination-port
   ```

   Use a destination port number from 1025 through 65536, but do not use 4500. If you do not specify a port number, the default destination port is 4565.

SEE ALSO

Understanding Service Sets | 8

Configuring Service Sets to be Applied to Services Interfaces | 10

Configuring Service Set Limitations | 15

Configuring System Logging for Service Sets | 24

Requesting for and Installing a Digital Certificates on Your Router

IN THIS SECTION

- Requesting a Digital Certificate—Manual Process | 743

A digital certificate is an electronic means for verifying your identity through a trusted third party, known as a certificate authority (CA). Alternatively, you can use a self-signed certificate to attest to your identity. The CA server you use can be owned and operated by an independent CA or by your own organization, in which case you become your own CA. If you use an independent CA, you must contact them for the addresses of their CA and certificate revocation list (CRL) servers (for obtaining certificates and CRLs) and for the information they require when submitting personal certificate requests. When you are your own CA, you determine this information yourself. The Public Key Infrastructure (PKI) provides an infrastructure for digital certificate management.

Requesting a Digital Certificate—Manual Process

To obtain digital certificates manually, you must configure a CA profile, generate a private-public key pair, create a local certificate, and load the certificates on the router. After loading the certificates, they can be referenced in your IPsec-VPN configuration.

This procedure shows how you can configure a CA profile:

1. Configure a CA profile:

   ```
   user@R2# set security pki ca-profile entrust ca-identity entrust enrollment url http://
   ca-1.example.com/cgi-bin/pkiclient.exe
   ```

   After you commit this configuration. The configuration on Router 2 must contain the following:

   ```
   [edit]
   security {
       pki {
           ca-profile entrust {
               ca-identity entrust;
               enrollment {
                   url http://ca-1.example.com/cgi-bin/pkiclient.exe;
               }
           }
       }
   }
   ```

2. Certificate revocation list (CRL) verification is enabled by default. You can optionally specify the Lightweight Access Directory (LDAP) server where the CA stores the CRL. The certificate typically includes a certificate distribution point (CDP), which contains information about how to retrieve the CRL for the certificate. The router uses this information to download the CRL automatically. In this example, the LDAP URL is specified, which overrides the location provided in the certificate:

   ```
   user@R2# set security pki ca-profile entrust revocation-check crl url ldap://10.157.90.185/
   o=juniper,c=uscertificateRevocationListbase
   ```

After you commit this configuration. The configuration on Router 2 must contain the following:

```
[edit]
security pki ca-profile entrust {
    revocation-check {
        crl {
            url ldap://10.157.90.185/o=juniper,c=uscertificateRevocationListbase;
        }
    }
}
```

3. After you configure the CA profile, request a CA certificate from the trusted CA. In this example, the certificate is enrolled online and installed into the router automatically.

```
user@R2> request security pki ca-certificate enroll ca-profile entrust
Received following certificates:
  Certificate: C=us, O=juniper
    Fingerprint: 00:8e:6f:58:dd:68:bf:25:0a:e3:f9:17:70:d6:61:f3:53:a7:79:10
  Certificate: C=us, O=juniper, CN=First Officer
    Fingerprint: bc:78:87:9b:a7:91:13:20:71:db:ac:b5:56:71:42:ad:1a:b6:46:17
  Certificate: C=us, O=juniper, CN=First Officer
    Fingerprint: 46:71:15:34:f0:a6:41:76:65:81:33:4f:68:47:c4:df:78:b8:e3:3f
Do you want to load the above CA certificate ? [yes,no] (no) yes
```

NOTE: If you obtain the CA certificate directly from the CA (for example, as an e-mail attachment or website download), you can install it with the `request security pki ca-certificate load` command.

4. Next, you must generate a private-public key pair before you can create a local certificate.

```
user@R2> request security pki generate-key-pair certificate-id local-entrust2
Generated key pair local-entrust2, key size 1024 bits
```

When the key pair is available, generate a local certificate request and send it to the CA for processing.

```
user@R2> request security pki generate-certificate-request
certificate-id local-entrust2 domain-name router2.example.com
```

```
filename entrust-req2 subject cn=router2.example.com
Generated certificate request
-----BEGIN CERTIFICATE REQUEST-----
MIIBoTCCAQoCAQAwGjEYMBYGA1UEAxMPdHAxLmp1bmlwZXIubmV0MIGfMA0GCSqG
SIb3DQEBAQUAA4GNADCBiQKBgQCiUFklQws1Ud+AqN5DDxRs2kVyKEhh9qoVFnz+
Hz4c9vsy3B8ElwTJlkmIt2cB3yifB6zePd+6WYpf57Crwre7YqPkiXM31F6z3YjX
H+1BPNbCxNWYvyrnSyVYDbFj8o0Xyqog8ACDfVL2JBWrPNBYy7imq/K9soDBbAs6
5hZqqwIDAQABoEcwRQYJKoZIhvcNAQkOMTgwNjAOBgNVHQ8BAf8EBAMCB4AwJAYD
VR0RAQH/BBowGIIWdHAxLmVuZ2xhYi5qdW5pcGVyLm5ldDANBgkqhkiG9w0BAQQF
AAOBgQBc2rq1v5SOQXH7LCb/FdqAL8ZM6GoaN5d6cGwq4bB6a7UQFgtoH406gQ3G
3iH0Zfz4xMIBpJYuGd1dkqgvcDoH3AgTsLkfn7Wi3x5H2qeQVs9bvL4P5nvEZLND
EIMUHwteolZCiZ70fO9Fer9cXWHSQs1UtXtgPqQJy2xIeImLgw==
-----END CERTIFICATE REQUEST-----
Fingerprint:
0d:90:b8:d2:56:74:fc:84:59:62:b9:78:71:9c:e4:9c:54:ba:16:97 (sha1)
1b:08:d4:f7:90:f1:c4:39:08:c9:de:76:00:86:62:b8 (md5)
```

NOTE: You can request the creation and installation of a local certificate online with the `request security pki local-certificate enroll` command.

5. The trusted CA digitally signs the local certificate and returns it to you. Copy the certificate file into the router and load the certificate.

```
user@R2> request security pki local-certificate load filename /tmp/router2-cert
certificate-id local-entrust2
Local certificate local-entrust2 loaded successfully
```

NOTE: The name of the file sent to you by the CA might not match the name of the certificate identifier. However, the `certificate-id` name must always match the name of the key pair you generated for the router.

Example: IKE Dynamic SA Configuration with Digital Certificates

IN THIS SECTION

- Requirements | 746

- Overview | 746
- Configuration | 747
- Verification | 765

This example shows how to configure IKE dynamic SA with digital certificates and contains the following sections.

Requirements

This example uses the following hardware and software components:

- Four M Series, MX Series, or T Series routers with multiservices interfaces installed in them.
- Junos OS Release 9.4 or later.

Before you configure this example you must request a CA certificate, create a local certificate, and load these digital certificates into the router. For details, see "Requesting for and Installing a Digital Certificates on Your Router" on page 742

Overview

IN THIS SECTION

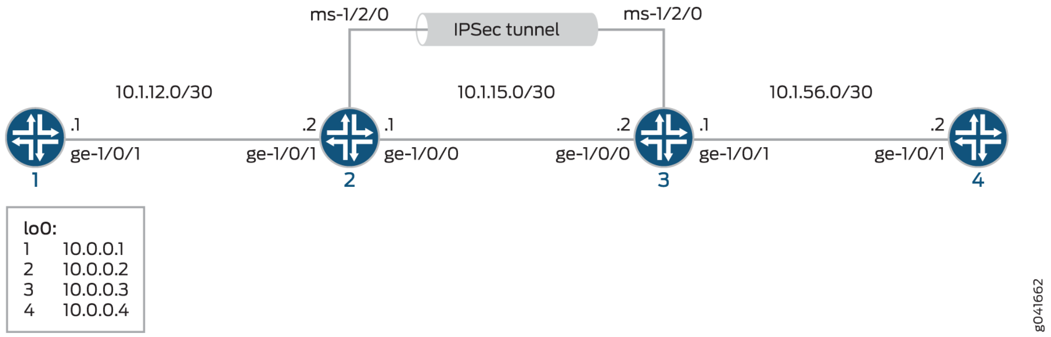
- Topology | 747

A security association (SA) is a simplex connection that enables two hosts to securely communicate with each other using IPsec. This example explains IKE dynamic SA configuration with digital certificates. The use of digital certificates provides additional security to your IKE tunnel. Using default values in the Services PIC, you do not need to configure an IPsec proposal or IPsec policy. However, you must configure an IKE proposal that specifies the use of digital certificates, reference the IKE proposal and local certificate in an IKE policy, and apply the CA profile to the service set.

Figure 40 on page 747 shows an IPsec topology containing a group of four routers. This configuration requires Routers 2 and 3 to establish an IKE-based IPsec tunnel by using digital certificates in place of preshared keys. Routers 1 and 4 provide basic connectivity and are used to verify that the IPsec tunnel is operational.

Topology

Figure 40: MS PIC IKE Dynamic SA Topology Diagram



Configuration

IN THIS SECTION

- Configuring Router 1 | 748
- Configuring Router 2 | 750
- Configuring Router 3 | 757
- Configuring Router 4 | 763

To configure IKE dynamic SA with digital certificates, perform these tasks:

NOTE: The interface types shown in this example are for indicative purpose only. For example, you can use `so-` interfaces instead of `ge-` and `sp-` instead of `ms-`.

Configuring Router 1

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI, at the [edit] hierarchy level, of Router 1.

```
set interfaces ge-0/0/0 description "to R2 ge-0/0/0"
set interfaces ge-0/0/0 unit 0 family inet address 10.1.12.2/30
set interfaces lo0 unit 0 family inet address 10.0.0.1/32
set routing-options router-id 10.0.0.1
set protocols ospf area 0.0.0.0 interface ge-0/0/0
set protocols ospf area 0.0.0.0 interface lo0.0
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure Router 1 for OSPF connectivity with Router 2:

1. Configure an Ethernet interface and the loopback interface.

   ```
   [edit interfaces]
   user@router1# set ge-0/0/0 description "to R2 ge-0/0/0"
   user@router1# set ge-0/0/0 unit 0 family inet address 10.1.12.2/30
   user@router1# set lo0 unit 0 family inet address 10.0.0.1/32
   ```

2. Specify the OSPF area and associate the interfaces with the OSPF area.

   ```
   [edit protocols]
   user@router1# set ospf area 0.0.0.0 interface ge-0/0/0.0
   user@router1# set ospf area 0.0.0.0 interface lo0.0
   ```

3. Configure the router ID.

```
[edit routing-options]
user@router1# set router-id 10.0.0.1
```

4. Commit the configuration.

```
[edit]
user@router1# commit
```

Results

From the configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols ospf`, and `show routing-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@router1# show interfaces
interfaces {
    ge-0/0/0 {
        description "To R2 ge-0/0/0";
        unit 0 {
            family inet {
                address 10.1.12.2/30;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.1/32;
            }
        }
    }
}
```

```
user@router1# show protocols ospf
protocols {
    ospf {
        area 0.0.0.0 {
            interface ge-0/0/0.0;
            interface lo0.0;
        }
    }
}
```

```
user@router1# show routing-options
routing-options  {
    router-id 10.0.0.1;
}
```

Configuring Router 2

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI, at the [edit] hierarchy level, of Router 2.

```
set interfaces ge-0/0/0 description "to R1 ge-0/0/0"
set interfaces ge-0/0/0 unit 0 family inet address 10.1.12.1/30
set interfaces ge-0/0/1 description "to R3 ge-0/0/1"
set interfaces ge-0/0/1 unit 0 family inet address 10.1.15.1/30
set interfaces ms-1/2/0 services-options syslog host local services info
set interfaces ms-1/2/0 unit 0 family inet
set interfaces ms-1/2/0 unit 1 family inet
set interfaces ms-1/2/0 unit 1 service-domain inside
set interfaces ms-1/2/0 unit 2 family inet
set interfaces ms-1/2/0 unit 2 service-domain outside
set interfaces lo0 unit 0 family inet address 10.0.0.2/32
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface ms-1/2/0.1
set routing-options router-id 10.0.0.2
set services ipsec-vpn rule rule-ike term term-ike then remote-gateway 10.1.15.2
set services ipsec-vpn rule rule-ike term term-ike then dynamic ike-policy ike-digital-
certificates
set services ipsec-vpn rule rule-ike term term-ike then dynamic ipsec-policy ipsec-demo-policy
set services ipsec-vpn rule match-direction input
set services ipsec-vpn ike proposal ike-demo-proposal authentication-method rsa-signatures
set services ipsec-vpn ike policy ike-digital-certificates proposals ike-demo-proposal
set services ipsec-vpn ike policy ike-digital-certificates local-id fqdn router2.example.com
set services ipsec-vpn ike policy ike-digital-certificates local-certificate local-entrust2
set services ipsec-vpn ike policy ike-digital-certificates remote-id fqdn router3.example.com
set services ipsec-vpn ipsec proposal ipsec-demo-proposal protocol esp
set services ipsec-vpn ipsec proposal ipsec-demo-proposal authentication-algorithm hmac-sha1-96
set services ipsec-vpn ipsec proposal ipsec-demo-proposal encryption-algorithm 3des-cbc
set services ipsec-vpn ipsec policy ipsec-demo-policy perfect-forward-secrecy keys group2
set services ipsec-vpn ipsec proposals ipsec-demo-proposal
set services ipsec-vpn establish-tunnels immediately
set services service-set demo-service-set next-hop-service inside-service-interface ms-1/2/0.1
set services service-set demo-service-set next-hop-service outside-service-interface ms-1/2/0.2
set services service-set demo-service-set ipsec-vpn-options trusted-ca entrust
set services service-set demo-service-set ipsec-vpn-options local-gateway 10.1.15.1
set services service-set demo-service-set ipsec-vpn-rules rule-ike
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure OSPF connectivity and IPsec tunnel parameters on Router 2:

1. Configure interface properties. In this step, you configure two Ethernet interfaces (ge-1/0/0 and ge-1/0/1), the loopback interface and a multiservices interface (ms-1/2/0).

   ```
   [edit interfaces]
   user@router2# set ge-0/0/0 description "to R1 ge-0/0/0"
   user@router2# set ge-0/0/0 unit 0 family inet address 10.1.12.1/30
   user@router2# set ge-0/0/1 description "to R3 ge-0/0/1"
   user@router2# set ge-0/0/1 unit 0 family inet address 10.1.15.1/30
   user@router2# set ms-1/2/0 services-options syslog host local services info
   user@router2# set ms-1/2/0 unit 0 family inet
   user@router2# set ms-1/2/0 unit 1 family inet
   user@router2# set ms-1/2/0 unit 1 service-domain inside
   user@router2# set ms-1/2/0 unit 2 family inet
   user@router2# set ms-1/2/0 unit 2 service-domain outside
   user@router2# set lo0 unit 0 family inet address 10.0.0.2/32
   ```

2. Specify the OSPF area and associate the interfaces with the OSPF area.

```
[edit protocols]
user@router2# set ospf area 0.0.0.0 interface ge-0/0/0.0
user@router2# set ospf area 0.0.0.0 interface lo0.0
user@router2# set ospf area 0.0.0.0 interface ms-1/2/0.1
```

3. Configure the router ID.

```
[edit routing-options]
user@router2# set router-ID 10.0.0.2
```

4. Configure an IKE proposal and policy. To enable an IKE proposal for digital certificates, include the `rsa-signatures` statement at the `[edit services ipsec-vpn ike proposal proposal-name authentication-method]` hierarchy level. To reference the local certificate in the IKE policy, include the `local-certificate` statement at the `[edit services ipsec-vpn ike policy policy-name]` hierarchy level. To identify the CA or RA in the service set, include the `trusted-ca` statement at the `[edit services service-set service-set-name ipsec-vpn-options]` hierarchy level.

NOTE: For information about creating and installing digital certificates, see "Requesting for and Installing a Digital Certificates on Your Router" on page 742

```
[edit services ipsec-vpn]
user@router2# set ike proposal ike-demo-proposal authentication-method rsa-signatures
user@router2# set ike policy ike-digital-certificates proposals ike-demo-proposal
user@router2# set ike policy ike-digital-certificates local-id fqdn router2.example.com
user@router2# set ike policy ike-digital-certificates local-certificate local-entrust2
user@router2# set ike policy ike-digital-certificates remote-id fqdn router3.example.com
```

5. Configure an IPsec proposal and policy. Also, set the `established-tunnels` knob to `immediately`.

```
[edit services ipsec-vpn]
user@router2# set ipsec proposal ipsec-demo-proposal protocol esp
user@router2# set ipsec proposal ipsec-demo-proposal authentication-algorithm hmac-sha1-96
user@router2# set ipsec proposal ipsec-demo-proposal encryption-algorithm 3des-cbc
user@router2# set ipsec policy ipsec-demo-policy perfect-forward-secrecy keys group2
```

```
user@router2# set ipsec proposals ipsec-demo-proposal
user@router2# set establish-tunnels immediately
```

6. Configure an IPsec rule.

```
[edit services ipsec-vpn]
user@router2# set rule rule-ike term term-ike then remote-gateway 10.1.15.2
user@router2# set rule rule-ike term term-ike then dynamic ike-policy ike-digital-certificates
user@router2# set rule rule-ike term term-ike then dynamic ipsec-policy ipsec-demo-policy
user@router2# set rule match-direction input
```

7. Configure a next-hop style service set, specify the local-gateway address, and associate the IPsec VPN rule with the service set.

```
[edit services]
user@router2# set service-set demo-service-set next-hop-service inside-service-interface
ms-1/2/0.1
user@router2# set service-set demo-service-set next-hop-service outside-service-interface
ms-1/2/0.2
user@router2# set service-set demo-service-set ipsec-vpn-options trusted-ca entrust
user@router2# set service-set demo-service-set ipsec-vpn-options local-gateway 10.1.15.1
user@router2# set service-set demo-service-set ipsec-vpn-rules rule-ike
```

8. Commit the configuration.

```
[edit]
user@router2# commit
```

Results

From the configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols ospf`, `show routing-options`, and `show services` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration

```
user@router2# show interfaces
interfaces {
    ge-0/0/0 {
        description "To R1 ge-0/0/0";
        unit 0 {
            family inet {
                address 10.1.12.1/30;
            }
        }
    }

    ge-0/0/1 {
        description "To R3 ge-0/0/1";
        unit 0 {
            family inet {
                address 10.1.15.1/30;
            }
        }
    }

    ms-1/2/0 {
        services-options {
            syslog {
                host local {
                    services info;
                }
            }
        }
        unit 0 {
            family inet;
        }
        unit 1 {
            family inet;
            service-domain inside;
        }
        unit 2 {
            family inet;
            service-domain outside;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.2/32;
            }
        }
    }
}
```

```
user@router2# show protocols ospf
protocols {
    ospf {
        area 0.0.0.0 {
            interface ge-0/0/0.0;
            interface lo0.0;
            interface ms-1/2/0.1;
        }
    }
}
```

```
user@router2# show routing-options
routing-options {
    router-id 10.0.0.2;
}
```

```
user@router2# show services
services {
    ipsec-vpn {
        rule rule-ike {
            term term-ike {
                then {
                    remote-gateway 10.1.15.2;
                    dynamic {
                        ike-policy ike-digital-certificates;
                        ipsec-policy ipsec-demo-policy
                    }
                }
            }
            match-direction input;
        }
        ike {
            proposal ike-demo-proposal {
                authentication-method rsa-signatures;
            }
            policy ike-digital-certificates {
                proposals ike-demo-proposal;
                local-id fqdn router2.example.com;
                local-certificate local-entrust2;
                remote-id fqdn router3.example.com;
            }
        }
        ipsec {
            proposal ipsec-demo-proposal {
                protocol esp;
                authentication-algorithm hmac-sha1-96;
                encryption-algorithm 3des-cbc;
            }
            policy demo-policy {
                perfect-forward-secrecy {
                    keys group2;
                }
                proposals ipsec-demo-proposal;
            }
            establish-tunnels immediately;
        }
        service-set service-set-dynamic-demo-service-set {
            next-hop-service {
                inside-service-interface ms-1/2/0.1;
                outside-service-interface ms-1/2/0.2;
            }
            ipsec-vpn-options {
                trusted-ca entrust;
                local-gateway 10.1.15.1;
            }
            ipsec-vpn-rules rule-ike;
        }
    }
}
```

Configuring Router 3

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI, at the [edit] hierarchy level, of Router 3.

```
set interfaces ge-0/0/0 description "to R4 ge-0/0/0"
set interfaces ge-0/0/0 unit 0 family inet address 10.1.56.1/30
set interfaces ge-0/0/1 description "to R2 ge-0/0/1"
set interfaces ge-0/0/1 unit 0 family inet address 10.1.15.2/30
set interfaces ms-1/2/0 services-options syslog host local services info
set interfaces ms-1/2/0 unit 0 family inet
set interfaces ms-1/2/0 unit 1 family inet
set interfaces ms-1/2/0 unit 1 service-domain inside
set interfaces ms-1/2/0 unit 2 family inet
set interfaces ms-1/2/0 unit 2 service-domain outside
set interfaces lo0 unit 0 family inet address 10.0.0.3/32
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface ms-1/2/0.1
set routing-options router-id 10.0.0.3
set services ipsec-vpn rule rule-ike term term-ike then remote-gateway 10.1.15.1
set services ipsec-vpn rule rule-ike term term-ike then dynamic ike-policy ike-digital-
certificates
set services ipsec-vpn rule rule-ike term term-ike then dynamic ipsec-policy ipsec-demo-policy
set services ipsec-vpn rule match-direction input
set services ipsec-vpn ike proposal ike-demo-proposal authentication-method rsa-signatures
set services ipsec-vpn ike policy ike-digital-certificates proposals ike-demo-proposal
set services ipsec-vpn ike policy ike-digital-certificates local-id fqdn router3.example.com
set services ipsec-vpn ike policy ike-digital-certificates local-certificate local-entrust3
set services ipsec-vpn ike policy ike-digital-certificates remote-id fqdn router2.example.com
set services ipsec-vpn ipsec proposal ipsec-demo-proposal protocol esp
set services ipsec-vpn ipsec proposal ipsec-demo-proposal authentication-algorithm hmac-sha1-96
set services ipsec-vpn ipsec proposal ipsec-demo-proposal encryption-algorithm 3des-cbc
set services ipsec-vpn ipsec policy ipsec-demo-policy perfect-forward-secrecy keys group2
set services ipsec-vpn ipsec proposals ipsec-demo-proposal
set services ipsec-vpn establish-tunnels immediately
set services service-set demo-service-set next-hop-service inside-service-interface ms-1/2/0.1
set services service-set demo-service-set next-hop-service outside-service-interface ms-1/2/0.2
set services service-set demo-service-set ipsec-vpn-options trusted-ca entrust
set services service-set demo-service-set ipsec-vpn-options local-gateway 10.1.15.2
set services service-set demo-service-set ipsec-vpn-rules rule-ike
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

> **NOTE**: If the IPsec peers do not have a symmetrical configuration containing all the necessary components, they cannot establish a peering relationship. You need to request a CA certificate, create a local certificate, load these digital certificates into the router, and reference them in your IPsec configuration. For information about digital certification, see "Requesting for and Installing a Digital Certificates on Your Router" on page 742

To configure OSPF connectivity and IPsec tunnel parameters on Router 3:

1. Configure interface properties. In this step, you configure two Ethernet interfaces (ge-1/0/0 and ge-1/0/1), the loopback interface, and a multiservices interface (ms-1/2/0).

    ```
    [edit interfaces]
    user@router3# set ge-0/0/0 description "to R4 ge-0/0/0"
    user@router3# set ge-0/0/0 unit 0 family inet address 10.1.56.1/30
    user@router3# set ge-0/0/1 description "to R2 ge-0/0/1"
    user@router3# set ge-0/0/1 unit 0 family inet address 10.1.15.2/30
    user@router3# set ms-1/2/0 services-options syslog host local services info
    user@router3# set ms-1/2/0 unit 0 family inet
    user@router3# set ms-1/2/0 unit 1 family inet
    user@router3# set ms-1/2/0 unit 1 service-domain inside
    user@router3# set ms-1/2/0 unit 2 family inet
    user@router3# set ms-1/2/0 unit 2 service-domain outside
    user@router3# set lo0 unit 0 family inet address 10.0.0.3/32
    ```

2. Specify the OSPF area, associate the interfaces with the OSPF area.

    ```
    [edit protocols]
    user@router3# set ospf area 0.0.0.0 interface ge-0/0/0.0
    user@router3# set ospf area 0.0.0.0 interface lo0.0
    user@router3# set ospf area 0.0.0.0 interface ms-1/2/0.1
    ```

3. Configure a router ID.

```
[edit routing-options]
user@router3# set router-id 10.0.0.3
```

4. Configure an IKE proposal and policy. To enable an IKE proposal for digital certificates, include the `rsa-signatures` statement at the `[edit services ipsec-vpn ike proposal proposal-name authentication-method]` hierarchy level. To reference the local certificate in the IKE policy, include the `local-certificate` statement at the `[edit services ipsec-vpn ike policy policy-name]` hierarchy level. To identify the CA or RA in the service set, include the `trusted-ca` statement at the `[edit services service-set service-set-name ipsec-vpn-options]` hierarchy level.

> **NOTE**: For information about creating and installing digital certificates, see "Requesting for and Installing a Digital Certificates on Your Router" on page 742

```
[edit services ipsec-vpn]
user@router3# set ike proposal ike-demo-proposal authentication-method rsa-signatures
user@router3# set ike policy ike-digital-certificates proposals ike-demo-proposal
user@router3# set ike policy ike-digital-certificates local-id fqdn router2.example.com
user@router3# set ike policy ike-digital-certificates local-certificate local-entrust2
user@router3# set ike policy ike-digital-certificates remote-id fqdn router3.example.com
```

5. Configure an IPsec proposal. Also, set the `established-tunnels` knob to `immediately`.

```
[edit services ipsec-vpn]
user@router3# set ipsec proposal ipsec-demo-proposal protocol esp
user@router3# set ipsec proposal ipsec-demo-proposal authentication-algorithm hmac-sha1-96
user@router3# set ipsec proposal ipsec-demo-proposal encryption-algorithm 3des-cbc
user@router3# set ipsec policy ipsec-demo-policy perfect-forward-secrecy keys group2
user@router3# set ipsec proposals ipsec-demo-proposal
user@router3# set establish-tunnels immediately
```

6. Configure an IPsec rule.

```
[edit services ipsec-vpn]
user@router3# set rule rule-ike term term-ike then remote-gateway 10.1.15.2
user@router3# set rule rule-ike term term-ike then dynamic ike-policy ike-digital-certificates
```

```
user@router3# set rule rule-ike term term-ike then dynamic ipsec-policy ipsec-demo-policy
user@router3# set rule match-direction input
```

7. Configure a next-hop style service set, specify the local-gateway address, and associate the IPsec VPN rule with the service set.

```
[edit services]
user@router3# set service-set demo-service-set next-hop-service inside-service-interface
ms-1/2/0.1
user@router3# set service-set demo-service-set next-hop-service outside-service-interface
ms-1/2/0.2
user@router3# set service-set demo-service-set ipsec-vpn-options trusted-ca entrust
user@router3# set service-set demo-service-set ipsec-vpn-options local-gateway 10.1.15.2
user@router3# set service-set demo-service-set ipsec-vpn-rules rule-ike
```

8. Commit the configuration.

```
[edit]
user@router3# commit
```

Results

From the configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols ospf`, `show routing-options`, and `show services` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration

```
user@router3# show interfaces
interfaces {
    ge-0/0/0 {
        description "To R4 ge-0/0/0";
        unit 0 {
            family inet {
                address 10.1.56.1/30;
            }
        }
    }
    ge-0/0/1 {
        description "To R2 ge-0/0/1";
        unit 0 {
            family inet {
                address 10.1.15.2/30;
            }
        }
    }
    ms-1/2/0 {
        services-options {
            syslog {
                host local {
                    services info;
                }
            }
        }
        unit 0 {
            family inet {
            }
            unit 1 {
                family inet;
                service-domain inside;
            }
            unit 2 {
                family inet;
                service-domain outside;
            }
        }
        lo0 {
            unit 0 {
                family inet {
                    address 10.0.0.3/32;
                }
            }
        }
    }
}
```

```
user@router3# show protocols ospf
protocols {
    ospf {
        area 0.0.0.0 {
            interface ge-0/0/0.0;
            interface lo0.0;
            interface ms-1/2/0.1;
        }
    }
}
```

```
user@router3# show routing-options
routing-options {
    router-id 10.0.0.3;
}
```

```
user@router3# show services
services {
    ipsec-vpn {
        rule rule-ike {
            term term-ike {
                then {
                    remote-gateway 10.1.15.1;
                    dynamic {
                        ike-policy ike-digital-certificates;
                        ipsec-policy ipsec-demo-policy
                    }
                }
            }
            match-direction input;
        }
        ike {
            proposal ike-demo-proposal {
                authentication-method rsa-signatures;
            }
            policy ike-digital-certificates {
                proposals ike-demo-proposal;
                local-id fqdn router3.example.com;
                local-certificate local-entrust3;
                remote-id fqdn router2.example.com;
            }
        }
        ipsec {
            proposal ipsec-demo-proposal {
                protocol esp;
                authentication-algorithm hmac-sha1-96;
                encryption-algorithm 3des-cbc;
            }
            policy demo-policy {
                perfect-forward-secrecy {
                    keys group2;
                }
                proposals ipsec-demo-proposal;
            }
            establish-tunnels immediately;
        }
        service-set service-set-dynamic-demo-service-set {
            next-hop-service {
                inside-service-interface ms-1/2/0.1;
                outside-service-interface ms-1/2/0.2;
            }
            ipsec-vpn-options {
                trusted-ca entrust;
                local-gateway 10.1.15.2;
            }
            ipsec-vpn-rules rule-ike;
        }
    }
}
```

Configuring Router 4

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI, at the [edit] hierarchy level, of Router 4.

```
set interfaces ge-0/0/0 description "to R3 ge-0/0/0"
set interfaces ge-0/0/0 unit 0 family inet address 10.1.56.2/30
set interfaces lo0 unit 0 family inet address 10.0.0.4/32
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0
set protocols ospf area 0.0.0.0 interface lo0.0
set routing-options router-id 10.0.0.4
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To set up OSPF connectivity with Router 4

1. Configure the interfaces. In this step, you configure an Ethernet interface (ge-1/0/1) and the loopback interface.

   ```
   [edit interfaces]
   user@router4# set ge-0/0/0 description "to R3 ge-0/0/0"
   user@router4# set ge-0/0/0 unit 0 family inet address 10.1.56.2/30
   user@router4# set lo0 unit 0 family inet address 10.0.0.4/32
   ```

2. Specify the OSPF area and associate the interfaces with the OSPF area.

   ```
   [edit protocols]
   user@router4# set ospf area 0.0.0.0 interface ge-0/0/0
   user@router4# set ospf area 0.0.0.0 interface lo0.0
   ```

3. Configure the router ID.

   ```
   [edit routing-options]
   user@router4# set router-id 10.0.0.4
   ```

Results

From the configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols ospf`, and `show routing-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration

```
user@router4# show interfaces
interfaces {
    ge-0/0/0 {
        description "To R3 ge-0/0/0";
        unit 0 {
            family inet {
```

```
                address 10.1.56.2/30;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.4/32;
            }
        }
    }
}
```

```
user@router4# show protocols ospf
protocols {
    ospf {
        area 0.0.0.0 {
            interface ge-0/0/0.0;
            interface lo0.0;
        }
    }
}
```

```
user@router4# show routing-options
routing-options {
    router-id 10.0.0.4;
}
```

Verification

IN THIS SECTION

- Verifying Your Work on Router 1 | **766**
- Verifying Your Work on Router 2 | **767**
- Verifying Your Work on Router 3 | **771**
- Verifying Your Work on Router 4 | **776**

Verifying Your Work on Router 1

Purpose

On Router 1, verify ping command to the so-0/0/0 interface on Router 4 to send traffic across the IPsec tunnel.

Action

From operational mode, enter `ping 10.1.56.2`.

```
user@router1>ping 10.1.56.2
PING 10.1.56.2 (10.1.56.2): 56 data bytes
64 bytes from 10.1.56.2: icmp_seq=0 ttl=254 time=1.351 ms
64 bytes from 10.1.56.2: icmp_seq=1 ttl=254 time=1.187 ms
64 bytes from 10.1.56.2: icmp_seq=2 ttl=254 time=1.172 ms
64 bytes from 10.1.56.2: icmp_seq=3 ttl=254 time=1.154 ms
64 bytes from 10.1.56.2: icmp_seq=4 ttl=254 time=1.156 ms
^C
--- 10.1.56.2 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.154/1.204/1.351/0.074 ms
```

If you ping the loopback address of Router 4, the operation succeeds because the address is part of the OSPF network configured on Router 4.

```
user@router1>ping 10.0.0.4
PING 10.0.0.4 (10.0.0.4): 56 data bytes
64 bytes from 10.0.0.4: icmp_seq=0 ttl=62 time=1.318 ms
64 bytes from 10.0.0.4: icmp_seq=1 ttl=62 time=1.084 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=62 time=3.260 ms
^C
--- 10.0.0.4 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.084/1.887/3.260/0.975 ms
```

Verifying Your Work on Router 2

Purpose

To verify that matched traffic is being diverted to the bidirectional IPsec tunnel, view the IPsec statistics:

Action

From operational mode, enter the `show services ipsec-vpn ipsec statistics`.

```
user@router2>show services ipsec-vpn ipsec statistics
PIC: sp-1/2/0, Service set: service-set-dynamic-demo-service-set
ESP Statistics:
Encrypted bytes: 162056
Decrypted bytes: 161896
Encrypted packets: 2215
Decrypted packets: 2216
AH Statistics:
Input bytes: 0
Output bytes: 0
Input packets: 0
Output packets: 0
Errors:
AH authentication failures: 0, Replay errors: 0
ESP authentication failures: 0, ESP decryption failures: 0
Bad headers: 0, Bad trailers: 0
```

To verify that the IKE SA negotiation is successful, issue the **show services ipsec-vpn ike security-associations** command:

From operational mode, enter the `show services ipsec-vpn ike security-associations`

```
user@router2> show services ipsec-vpn ike security-associations
Remote Address State Initiator cookie Responder cookie Exchange type
10.1.15.2 Matured d82610c59114fd37 ec4391f76783ef28 Main
```

To verify that the IPsec security association is active, issue the **show services ipsec-vpn ipsec security-associations detail** command. Notice that the SA contains the default settings inherent in the Services PIC, such as ESP for the protocol and HMAC-SHA1-96 for the authentication algorithm.

From operational mode, enter the `show services ipsec-vpn ipsec security-associations detail`

```
user@router2> show services ipsec-vpn ipsec security-associations detail
Service set: service-set-dynamic-demo-service-set
Rule: rule-ike, Term: term-ike, Tunnel index: 1
Local gateway: 10.1.15.1, Remote gateway: 10.1.15.2
IPsec inside interface: sp-1/2/0.1
Local identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
Remote identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
Direction: inbound, SPI: 857451461, AUX-SPI: 0
Mode: tunnel, Type: dynamic, State: Installed
Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
Soft lifetime: Expires in 9052 seconds
Hard lifetime: Expires in 9187 seconds
Anti-replay service: Enabled, Replay window size: 64
Direction: outbound, SPI: 1272330309, AUX-SPI: 0
Mode: tunnel, Type: dynamic, State: Installed
Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
Soft lifetime: Expires in 9052 seconds
Hard lifetime: Expires in 9187 seconds
Anti-replay service: Enabled, Replay window size: 64
```

To display the digital certificates that are used to establish the IPsec tunnel, issue the show services ipsec-vpn certificates command:

From operational mode, enter the `show services ipsec-vpn certificates`

```
user@router2> show services ipsec-vpn certificates
Service set: service-set-dynamic-demo-service-set, Total entries: 3
Certificate cache entry: 3
Flags: Non-root Trusted
Issued to: router3.example.com, Issued by: juniper
Alternate subject: router3.example.com
Validity:
Not before: 2005 Nov 21st, 23:33:58 GMT
Not after: 2008 Nov 22nd, 00:03:58 GMT
Certificate cache entry: 2
Flags: Non-root Trusted
Issued to: router2.example.com, Issued by: juniper
Alternate subject: router2.example.com
Validity:
Not before: 2005 Nov 21st, 23:28:22 GMT
Not after: 2008 Nov 21st, 23:58:22 GMT
Certificate cache entry: 1
Flags: Root Trusted
Issued to: juniper, Issued by: juniper
Validity:
Not before: 2005 Oct 18th, 23:54:22 GMT
Not after: 2025 Oct 19th, 00:24:22 GMT
```

To display the CA certificate, issue the show security pki ca-certificate detail command. Notice that there are three separate certificates: one for certificate signing, one for key encipherment, and one for the CA's digital signature.

From operational mode, enter the `show security pki ca-certificate detail`

```
user@router2> show security pki ca-certificate detail
Certificate identifier: entrust
Certificate version: 3
Serial number: 4355 9235
Issuer:
Organization: juniper, Country: us
Subject:
Organization: juniper, Country: us
Validity:
Not before: 2005 Oct 18th, 23:54:22 GMT
Not after: 2025 Oct 19th, 00:24:22 GMT
Public key algorithm: rsaEncryption(1024 bits)
cb:9e:2d:c0:70:f8:ea:3c:f2:b5:f0:02:48:87:dc:68:99:a3:57:4f
0e:b9:98:0b:95:47:0d:1f:97:7c:53:17:dd:1a:f8:da:e5:08:d1:1c
78:68:1f:2f:72:9f:a2:cf:81:e3:ce:c5:56:89:ce:f0:97:93:fa:36
19:3e:18:7d:8c:9d:21:fe:1f:c3:87:8d:b3:5d:f3:03:66:9d:16:a7
bf:18:3f:f0:7a:80:f0:62:50:43:83:4f:0e:d7:c6:42:48:c0:8a:b2
c7:46:30:38:df:9b:dc:bc:b5:08:7a:f3:cd:64:db:2b:71:67:fe:d8
04:47:08:07:de:17:23:13
Signature algorithm: sha1WithRSAEncryption
Fingerprint:
00:8e:6f:58:dd:68:bf:25:0a:e3:f9:17:70:d6:61:f3:53:a7:79:10 (sha1)
71:6f:6a:76:17:9b:d6:2a:e7:5a:72:97:82:6d:26:86 (md5)
Distribution CRL:
C=us, O=juniper, CN=CRL1
http://CA-1/CRL/juniper_us_crlfile.crl
Use for key: CRL signing, Certificate signing
Certificate identifier: entrust
Certificate version: 3
Serial number: 4355 925c
Issuer:
Organization: juniper, Country: us
Subject:
Organization: juniper, Country: us, Common name: First Officer
Validity:
Not before: 2005 Oct 18th, 23:55:59 GMT
Not after: 2008 Oct 19th, 00:25:59 GMT
Public key algorithm: rsaEncryption(1024 bits)
c0:a4:21:32:95:0a:cd:ec:12:03:d1:a2:89:71:8e:ce:4e:a6:f9:2f
1a:9a:13:8c:f6:a0:3d:c9:bd:9d:c2:a0:41:77:99:1b:1e:ed:5b:80
34:46:f8:5b:28:34:38:2e:91:7d:4e:ad:14:86:78:67:e7:02:1d:2e
19:11:b7:fa:0d:ba:64:20:e1:28:4e:3e:bb:6e:64:dc:cd:b1:b4:7a
ca:8f:47:dd:40:69:c2:35:95:ce:b8:85:56:d7:0f:2d:04:4d:5d:d8
42:e1:4f:6b:bf:38:c0:45:1e:9e:f0:b4:7f:74:6f:e9:70:fd:4a:78
da:eb:10:27:bd:46:34:33
Signature algorithm: sha1WithRSAEncryption
Fingerprint:
bc:78:87:9b:a7:91:13:20:71:db:ac:b5:56:71:42:ad:1a:b6:46:17 (sha1)
23:79:40:c9:6d:a6:f0:ca:e0:13:30:d4:29:6f:86:79 (md5)
Distribution CRL:
C=us, O=juniper, CN=CRL1
http://CA-1/CRL/juniper_us_crlfile.crl
Use for key: Key encipherment
Certificate identifier: entrust
Certificate version: 3
Serial number: 4355 925b
Issuer:
Organization: juniper, Country: us
Subject:
Organization: juniper, Country: us, Common name: First Officer
Validity:
Not before: 2005 Oct 18th, 23:55:59 GMT
Not after: 2008 Oct 19th, 00:25:59 GMT
Public key algorithm: rsaEncryption(1024 bits)
ea:75:c4:f3:58:08:ea:65:5c:7e:b3:de:63:0a:cf:cf:ec:9a:82:e2
d7:e8:b9:2f:bd:4b:cd:86:2f:f1:dd:d8:a2:95:af:ab:51:a5:49:4e
00:10:c6:25:ff:b5:49:6a:99:64:74:69:e5:8c:23:5b:b4:70:62:8e
e4:f9:a2:28:d4:54:e2:0b:1f:50:a2:92:cf:6c:8f:ae:10:d4:69:3c
90:e2:1f:04:ea:ac:05:9b:3a:93:74:d0:59:24:e9:d2:9d:c2:ef:22
b9:32:c7:2c:29:4f:91:cb:5a:26:fe:1d:c0:36:dc:f4:9c:8b:f5:26
af:44:bf:53:aa:d4:5f:67
```

```
Signature algorithm: sha1WithRSAEncryption
Fingerprint:
46:71:15:34:f0:a6:41:76:65:81:33:4f:68:47:c4:df:78:b8:e3:3f (sha1)
ee:cc:c7:f4:5d:ac:65:33:0a:55:db:59:72:2c:dd:16 (md5)
Distribution CRL:
C=us, O=juniper, CN=CRL1
http://CA-1/CRL/juniper_us_crlfile.crl
Use for key: Digital signature
```

To display the local certificate request, issue the show security pki certificate-request command:

From operational mode, enter the `show security pki certificate-request`

```
user@router2> show security pki certificate-request
Certificate identifier: local-entrust2
Issued to: router2.example.com
Public key algorithm: rsaEncryption(1024 bits)
Public key verification status: Passed
```

To display the local certificate, issue the show security pki local-certificate command:

From operational mode, enter the `show security pki local-certificate`

```
user@router2> show security pki local-certificate
Certificate identifier: local-entrust2
Issued to: router2.example.com, Issued by: juniper
Validity:
Not before: 2005 Nov 21st, 23:28:22 GMT
Not after: 2008 Nov 21st, 23:58:22 GMT
Public key algorithm: rsaEncryption(1024 bits)
Public key verification status: Passed
```

Verifying Your Work on Router 3

Purpose

To verify that matched traffic is being diverted to the bidirectional IPsec tunnel, view the IPsec statistics:

Action

From operational mode, enter the `show services ipsec-vpn ipsec statistics`.

```
user@router3>show services ipsec-vpn ipsec statistics
PIC: sp-1/2/0, Service set: service-set-dynamic-demo-service-set
ESP Statistics:
Encrypted bytes: 161896
Decrypted bytes: 162056
Encrypted packets: 2216
Decrypted packets: 2215
AH Statistics:
Input bytes: 0
Output bytes: 0
Input packets: 0
Output packets: 0
Errors:
AH authentication failures: 0, Replay errors: 0
ESP authentication failures: 0, ESP decryption failures: 0
Bad headers: 0, Bad trailers: 0
```

To verify that the IKE SA negotiation is successful, issue the show services ipsec-vpn ike security-associations command. To be successful, the SA on Router 3 must contain the same settings you specified on Router 2.

From operational mode, enter the `show services ipsec-vpn ike security-associations`.

```
user@router3>show services ipsec-vpn ike security-associations
Remote Address State Initiator cookie Responder cookie Exchange type
10.1.15.1 Matured d82610c59114fd37 ec4391f76783ef28 Main
```

To verify that the IPsec SA is active, issue the show services ipsec-vpn ipsec security-associations detail command. To be successful, the SA on Router 3 must contain the same settings you specified on Router 2.

From operational mode, enter the `show services ipsec-vpn ipsec security-associations detail`.

```
user@router3>show services ipsec-vpn ipsec security-associations detail
Service set: service-set-dynamic-demo-service-set
Rule: rule-ike, Term: term-ike, Tunnel index: 1
Local gateway: 10.1.15.2, Remote gateway: 10.1.15.1
```

```
IPsec inside interface: sp-1/2/0.1
Local identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
Remote identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
Direction: inbound, SPI: 1272330309, AUX-SPI: 0
Mode: tunnel, Type: dynamic, State: Installed
Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
Soft lifetime: Expires in 7219 seconds
Hard lifetime: Expires in 7309 seconds
Anti-replay service: Enabled, Replay window size: 64
Direction: outbound, SPI: 857451461, AUX-SPI: 0
Mode: tunnel, Type: dynamic, State: Installed
Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
Soft lifetime: Expires in 7219 seconds
Hard lifetime: Expires in 7309 seconds
Anti-replay service: Enabled, Replay window size: 64
```

To display the digital certificates that are used to establish the IPsec tunnel, issue the show services ipsec-vpn certificates command:

From operational mode, enter the `show services ipsec-vpn certificates`.

```
user@router3>show services ipsec-vpn certificates
Service set: service-set-dynamic-demo-service-set, Total entries: 3
Certificate cache entry: 3
Flags: Non-root Trusted
Issued to: router3.example.com, Issued by: juniper
Alternate subject: router3.example.com
Validity:
Not before: 2005 Nov 21st, 23:33:58 GMT
Not after: 2008 Nov 22nd, 00:03:58 GMT
Certificate cache entry: 2
Flags: Non-root Trusted
Issued to: router2.example.com, Issued by: juniper
Alternate subject: router2.example.com
Validity:
Not before: 2005 Nov 21st, 23:28:22 GMT
Not after: 2008 Nov 21st, 23:58:22 GMT
Certificate cache entry: 1
Flags: Root Trusted
Issued to: juniper, Issued by: juniper
Validity:
```

```
Not before: 2005 Oct 18th, 23:54:22 GMT
Not after: 2025 Oct 19th, 00:24:22 GMT
```

To display the CA certificate, issue the show security pki ca-certificate detail command. Notice that there are three separate certificates: one for certificate signing, one for key encipherment, and one for the CA's digital signature.

From operational mode, enter the `show security pki ca-certificate detail`.

```
user@router3>show security pki ca-certificate detail
Certificate identifier: entrust
Certificate version: 3
Serial number: 4355 9235
Issuer:
Organization: juniper, Country: us
Subject:
Organization: juniper, Country: us
Validity:
Not before: 2005 Oct 18th, 23:54:22 GMT
Not after: 2025 Oct 19th, 00:24:22 GMT
Public key algorithm: rsaEncryption(1024 bits)
cb:9e:2d:c0:70:f8:ea:3c:f2:b5:f0:02:48:87:dc:68:99:a3:57:4f
0e:b9:98:0b:95:47:0d:1f:97:7c:53:17:dd:1a:f8:da:e5:08:d1:1c
78:68:1f:2f:72:9f:a2:cf:81:e3:ce:c5:56:89:ce:f0:97:93:fa:36
19:3e:18:7d:8c:9d:21:fe:1f:c3:87:8d:b3:5d:f3:03:66:9d:16:a7
bf:18:3f:f0:7a:80:f0:62:50:43:83:4f:0e:d7:c6:42:48:c0:8a:b2
c7:46:30:38:df:9b:dc:bc:b5:08:7a:f3:cd:64:db:2b:71:67:fe:d8
04:47:08:07:de:17:23:13
Signature algorithm: sha1WithRSAEncryption
Fingerprint:
00:8e:6f:58:dd:68:bf:25:0a:e3:f9:17:70:d6:61:f3:53:a7:79:10 (sha1)
71:6f:6a:76:17:9b:d6:2a:e7:5a:72:97:82:6d:26:86 (md5)
Distribution CRL:
C=us, O=juniper, CN=CRL1
http://CA-1/CRL/juniper_us_crlfile.crl
Use for key: CRL signing, Certificate signing
Certificate identifier: entrust
Certificate version: 3
Serial number: 4355 925c
Issuer:
Organization: juniper, Country: us
Subject:
```

```
Organization: juniper, Country: us, Common name: First Officer
Validity:
Not before: 2005 Oct 18th, 23:55:59 GMT
Not after: 2008 Oct 19th, 00:25:59 GMT
Public key algorithm: rsaEncryption(1024 bits)
c0:a4:21:32:95:0a:cd:ec:12:03:d1:a2:89:71:8e:ce:4e:a6:f9:2f
1a:9a:13:8c:f6:a0:3d:c9:bd:9d:c2:a0:41:77:99:1b:1e:ed:5b:80
34:46:f8:5b:28:34:38:2e:91:7d:4e:ad:14:86:78:67:e7:02:1d:2e
19:11:b7:fa:0d:ba:64:20:e1:28:4e:3e:bb:6e:64:dc:cd:b1:b4:7a
ca:8f:47:dd:40:69:c2:35:95:ce:b8:85:56:d7:0f:2d:04:4d:5d:d8
42:e1:4f:6b:bf:38:c0:45:1e:9e:f0:b4:7f:74:6f:e9:70:fd:4a:78
da:eb:10:27:bd:46:34:33
Signature algorithm: sha1WithRSAEncryption
Fingerprint:
bc:78:87:9b:a7:91:13:20:71:db:ac:b5:56:71:42:ad:1a:b6:46:17 (sha1)
23:79:40:c9:6d:a6:f0:ca:e0:13:30:d4:29:6f:86:79 (md5)
Distribution CRL:
C=us, O=juniper, CN=CRL1
http://CA-1/CRL/juniper_us_crlfile.crl
Use for key: Key encipherment
Certificate identifier: entrust
Certificate version: 3
Serial number: 4355 925b
Issuer:
Organization: juniper, Country: us
Subject:
Organization: juniper, Country: us, Common name: First Officer
Validity:
Not before: 2005 Oct 18th, 23:55:59 GMT
Not after: 2008 Oct 19th, 00:25:59 GMT
Public key algorithm: rsaEncryption(1024 bits)
ea:75:c4:f3:58:08:ea:65:5c:7e:b3:de:63:0a:cf:cf:ec:9a:82:e2
d7:e8:b9:2f:bd:4b:cd:86:2f:f1:dd:d8:a2:95:af:ab:51:a5:49:4e
00:10:c6:25:ff:b5:49:6a:99:64:74:69:e5:8c:23:5b:b4:70:62:8e
e4:f9:a2:28:d4:54:e2:0b:1f:50:a2:92:cf:6c:8f:ae:10:d4:69:3c
90:e2:1f:04:ea:ac:05:9b:3a:93:74:d0:59:24:e9:d2:9d:c2:ef:22
b9:32:c7:2c:29:4f:91:cb:5a:26:fe:1d:c0:36:dc:f4:9c:8b:f5:26
af:44:bf:53:aa:d4:5f:67
Signature algorithm: sha1WithRSAEncryption
Fingerprint:
46:71:15:34:f0:a6:41:76:65:81:33:4f:68:47:c4:df:78:b8:e3:3f (sha1)
ee:cc:c7:f4:5d:ac:65:33:0a:55:db:59:72:2c:dd:16 (md5)
Distribution CRL:
```

```
C=us, O=juniper, CN=CRL1
http://CA-1/CRL/juniper_us_crlfile.crl
Use for key: Digital signature
```

To display the local certificate request, issue the show security pki certificate-request command:

From operational mode, enter the `show security pki certificate-request`.

```
user@router3>show security pki certificate-request
Certificate identifier: local-entrust3
Issued to: router3.example.com
Public key algorithm: rsaEncryption(1024 bits)
Public key verification status: Passed
```

To display the local certificate, issue the show security pki local-certificate command:

From operational mode, enter the `show security pki local-certificate`.

```
user@router3>show security pki local-certificate
Certificate identifier: local-entrust3
Issued to: router3.example.com, Issued by: juniper
Validity:
Not before: 2005 Nov 21st, 23:33:58 GMT
Not after: 2008 Nov 22nd, 00:03:58 GMT
Public key algorithm: rsaEncryption(1024 bits)
Public key verification status: Passed
```

Verifying Your Work on Router 4

Purpose

On Router 4, issue a ping command to the so-0/0/0 interface on Router 1 to send traffic across the IPsec tunnel.

Action

From operational mode, enter `ping 10.1.12.2`.

```
user@router4>ping 10.1.12.2
PING 10.1.12.2 (10.1.12.2): 56 data bytes
```

```
64 bytes from 10.1.12.2: icmp_seq=0 ttl=254 time=1.350 ms
64 bytes from 10.1.12.2: icmp_seq=1 ttl=254 time=1.161 ms
64 bytes from 10.1.12.2: icmp_seq=2 ttl=254 time=1.124 ms
64 bytes from 10.1.12.2: icmp_seq=5 ttl=254 time=1.116 ms
^C
--- 10.1.12.2 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.116/1.172/1.350/0.081 ms
```

The final way you can confirm that traffic travels over the IPsec tunnel is by issuing the traceroute command to the so-0/0/0 interface on Router 1. Notice that the physical interface between Routers 2 and 3 is not referenced in the path; traffic enters the IPsec tunnel through the adaptive services IPsec inside interface on Router 3, passes through the loopback interface on Router 2, and ends at the so-0/0/0 interface on Router 1.

From operational mode, enter the `traceroute 10.1.12.2`.

```
user@router4>traceroute 10.1.12.2
traceroute to 10.1.12.2 (10.1.12.2), 30 hops max, 40 byte packets
1 10.1.15.2 (10.1.15.2) 0.987 ms 0.630 ms 0.563 ms
2 10.0.0.2 (10.0.0.2) 1.194 ms 1.058 ms 1.033 ms
3 10.1.12.2 (10.1.12.2) 1.073 ms 0.949 ms 0.932 ms
```

SEE ALSO

- Understanding Junos VPN Site Secure | 623
- Configuring Security Associations | 645
- Configuring IKE Proposals | 675
- Configuring IKE Policies | 681
- Example: Configuring IKE Dynamic SAs | 699
- Example: Configuring Manual SAs | 653
- Requesting for and Installing a Digital Certificates on Your Router | 742

Configuring Junos VPN Site Secure or IPSec VPN

IPsec VPN is supported on all MX Series routers with MS-MICs, MS-MPCs, or MS-DPCs.

On M Series and T Series routers, IPsec VPN is supported with Multiservices 100 PICs, Multiservices 400 PICs, and Multiservices 500 PICs.

MS-MICs and MS-MPCs are supported from Junos OS Release 13.2 and later. MS-MICs and MS-MPCs support all features that are supported by MS-DPCs and MS-PICs except for authentication header protocol (ah), encapsulating security payload protocol (esp), and bundle (ah and esp protocol) protocol for a dynamic or manual security association and flowless IPsec service.

NAT traversal (NAT-T) is supported for IKEv1 and IKEv2 from Junos OS Release 17.4R1 onwards. NAT-T is enabled by default. You can specify the UDP encapsulation and decapsulation for IKE and ESP packets using the configuration `disable-natt` at the `[edit services ipsec-vpn]` hierarchy levels.

SEE ALSO

Configuring Security Associations | 645

Example: Configuring Junos VPN Site Secure on MS-MIC and MS-MPC

IN THIS SECTION

- Requirements | 778
- Overview | 779
- Configuration | 780
- Verification | 791

NOTE: You can follow the same procedure and use the same configuration given in this example, to configure Junos VPN Site Secure (previously known as IPsec features) on MS-MPCs.

This example contains the following sections:

Requirements

This example uses the following hardware and software components:

- Two MX Series routers with MS-MICs
- Junos OS Release 13.2 or later

Overview

Junos OS Release 13.2, extends support for Junos VPN Site Secure (formerly known as IPsec features) to the newly-introduced Multiservices MIC and MPC (MS-MIC and MS-MPC) on MX Series routers. The Junos OS extension-provider packages come preinstalled and preconfigured on the MS-MIC and MS-MPC.

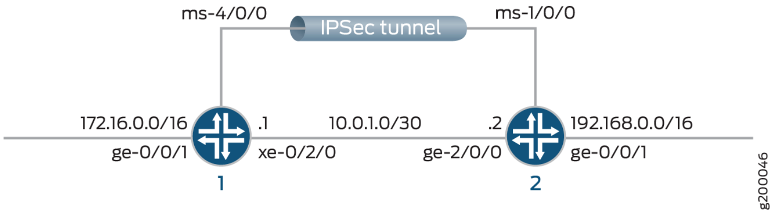
The following Junos VPN Site Secure features are supported on the MS-MIC and MS-MPC in Release 13.2:

- Dynamic End Points (DEP)
- Encapsulating Security Payload (ESP) protocol
- Dead Peer Detection (DPD) trigger messages
- Sequence Number Rollover notifications
- Static IPsec tunnels with next-hop-style and interface-style service sets

However, in Junos OS Release 13.2, the Junos VPN Site Secure support on the MS-MIC and MS-MPC is limited to IPv4 traffic. Passive module tunneling is not supported on MS-MICs and MS-MPCs.

Figure 41 on page 779 shows the IPsec VPN tunnel topology.

Figure 41: IPsec VPN Tunnel Topology



This example shows configuration of two routers, Router 1 and Router 2, that have an IPsec VPN tunnel configured between them.

While configuring the routers, note the following points:

- The IP address you configure for `source-address` under the `[edit services ipsec-vpn rule` *`name`* `term` *`term`* `from]` hierarchy level on Router 1 must be the same as the IP address you configure for `destination-address` under the same hierarchy on Router 2, and vice versa.

- The IP address of the `remote-gateway` you configure under the `[edit services ipsec-vpn rule` *`name`* `term` *`term`* `then]` hierarchy level should match the IP address of the `local-gateway` you configure under the `[edit services service-set` *`name`* `ipsec-vpn-options]` hierarchy level of Router 2, and vice versa.

Configuration

IN THIS SECTION

- CLI Quick Configuration | 780
- Configuring Router 1 | 782
- Configuring Router 2 | 787

This section contains:

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the [edit] hierarchy level.

Configuring Interfaces on Router 1

```
set interfaces ms-4/0/0 unit 0 family inet
set interfaces ms-4/0/0 unit 1 family inet
set interfaces ms-4/0/0 unit 1 family inet6
set interfaces ms-4/0/0 unit 1 service-domain inside
set interfaces ms-4/0/0 unit 2 family inet
set interfaces ms-4/0/0 unit 2 family inet6
set interfaces ms-4/0/0 unit 2 service-domain outside
set interfaces xe-0/2/0 unit 0 family inet address 10.0.1.1/30
```

Configuring IPsec VPN Service on Router 1

```
set services ipsec-vpn rule vpn_rule_ms_4_0_01 term term11 from source-address 172.16.0.0/16
set services ipsec-vpn rule vpn_rule_ms_4_0_01 term term11 from destination-address
192.168.0.0/16
set services ipsec-vpn rule vpn_rule_ms_4_0_01 term term11 then remote-gateway 10.0.1.2
set services ipsec-vpn rule vpn_rule_ms_4_0_01 term term11 then dynamic ike-policy
ike_policy_ms_4_0_0
set services ipsec-vpn rule vpn_rule_ms_4_0_01 term term11 then dynamic ipsec-policy
ipsec_policy_ms_4_0_0
set services ipsec-vpn rule vpn_rule_ms_4_0_01 term term11 then anti-replay-window-size 4096
set services ipsec-vpn rule vpn_rule_ms_4_0_01 match-direction input
set services ipsec-vpn ipsec proposal ipsec_proposal_ms_4_0_0 protocol esp
set services ipsec-vpn ipsec proposal ipsec_proposal_ms_4_0_0 authentication-algorithm hmac-
sha1-96
set services ipsec-vpn ipsec proposal ipsec_proposal_ms_4_0_0 encryption-algorithm 3des-cbc
set services ipsec-vpn ipsec policy ipsec_policy_ms_4_0_0 perfect-forward-secrecy keys group2
set services ipsec-vpn ipsec policy ipsec_policy_ms_4_0_0 proposals ipsec_proposal_ms_4_0_0
set services ipsec-vpn ike proposal ike_proposal_ms_4_0_0 authentication-method pre-shared-keys
set services ipsec-vpn ike proposal ike_proposal_ms_4_0_0 dh-group group2
set services ipsec-vpn ike policy ike_policy_ms_4_0_0 version 2
set services ipsec-vpn ike policy ike_policy_ms_4_0_0 proposals ike_proposal_ms_4_0_0
set services ipsec-vpn ike policy ike_policy_ms_4_0_0 pre-shared-key ascii-text secret-data
```

Configuring a Service Set on Router 1

```
set services service-set ipsec_ss_ms_4_0_01 next-hop-service inside-service-interface ms-4/0/0.1
set services service-set ipsec_ss_ms_4_0_01 next-hop-service outside-service-interface ms-4/0/0.2
set services service-set ipsec_ss_ms_4_0_01 ipsec-vpn-options local-gateway 10.0.1.1
set services service-set ipsec_ss_ms_4_0_01 ipsec-vpn-rules vpn_rule_ms_4_0_01
```

Configuring Routing Options on Router 1

```
set routing-options static route 192.168.0.0/16 next-hop ms-4/0/0.1
```

Configuring Interfaces on Router 2

```
set interfaces ms-1/0/0 unit 0 family inet
set interfaces ms-1/0/0 unit 1 family inet
set interfaces ms-1/0/0 unit 1 family inet6
set interfaces ms-1/0/0 unit 1 service-domain inside
set interfaces ms-1/0/0 unit 2 family inet
set interfaces ms-1/0/0 unit 2 family inet6
set interfaces ms-1/0/0 unit 2 service-domain outside
set interfaces ge-2/0/0 unit 0 family inet address 10.0.1.2/30
```

Configuring IPsec VPN Service on Router 2

```
set services ipsec-vpn rule vpn_rule_ms_5_2_01 term term11 from source-address 192.168.0.0/16
set services ipsec-vpn rule vpn_rule_ms_5_2_01 term term11 from destination-address 172.16.0.0/16
set services ipsec-vpn rule vpn_rule_ms_5_2_01 term term11 then remote-gateway 10.0.1.1
set services ipsec-vpn rule vpn_rule_ms_5_2_01 term term11 then dynamic ike-policy
ike_policy_ms_5_2_0
set services ipsec-vpn rule vpn_rule_ms_5_2_01 term term11 then dynamic ipsec-policy
ipsec_policy_ms_5_2_0
set services ipsec-vpn rule vpn_rule_ms_5_2_01 term term11 then anti-replay-window-size 4096
set services ipsec-vpn rule vpn_rule_ms_5_2_01 match-direction input
set services ipsec-vpn ipsec proposal ipsec_proposal_ms_5_2_0 protocol esp
set services ipsec-vpn ipsec proposal ipsec_proposal_ms_5_2_0 authentication-algorithm hmac-
sha1-96
set services ipsec-vpn ipsec proposal ipsec_proposal_ms_5_2_0 encryption-algorithm 3des-cbc
set services ipsec-vpn ipsec policy ipsec_policy_ms_5_2_0 perfect-forward-secrecy keys group2
set services ipsec-vpn ipsec policy ipsec_policy_ms_5_2_0 proposals ipsec_proposal_ms_5_2_0
set services ipsec-vpn ike proposal ike_proposal_ms_5_2_0 authentication-method pre-shared-keys
set services ipsec-vpn ike proposal ike_proposal_ms_5_2_0 dh-group group2
set services ipsec-vpn ike policy ike_policy_ms_5_2_0 version 2
set services ipsec-vpn ike policy ike_policy_ms_5_2_0 proposals ike_proposal_ms_5_2_0
set services ipsec-vpn ike policy ike_policy_ms_5_2_0 pre-shared-key ascii-text secret-data
set services ipsec-vpn establish-tunnels immediately
```

Configuring a Service Set on Router 2

```
set services service-set ipsec_ss_ms_5_2_01 next-hop-service inside-service-interface ms-1/0/0.1
set services service-set ipsec_ss_ms_5_2_01 next-hop-service outside-service-interface ms-1/0/0.2
set services service-set ipsec_ss_ms_5_2_01 ipsec-vpn-options local-gateway 10.0.1.2
set services service-set ipsec_ss_ms_5_2_01 ipsec-vpn-rules vpn_rule_ms_5_2_01
```

Configuring Routing Options on Router 2

```
set routing-options static route 172.16.0.0/16 next-hop ms-1/0/0.1
```

Configuring Router 1

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

NOTE: Starting with Release 13.2, the Junos OS extension-provider packages come preinstalled on multiservices MICs and MPCs (MS-MICs and MS-MPCs). The `adaptive-services` configuration at the `[edit chassis fpc *number* pic *number*]` hierarchy level is preconfigured on these cards.

1. Configure the interface properties such as family, service-domain, and unit.

```
user@router1# set interfaces ms-4/0/0 unit 0 family inet
user@router1# set interfaces ms-4/0/0 unit 1 family inet
user@router1# set interfaces ms-4/0/0 unit 1 family inet6
user@router1# set interfaces ms-4/0/0 unit 1 service-domain inside
user@router1# set interfaces ms-4/0/0 unit 2 family inet
user@router1# set interfaces ms-4/0/0 unit 2 family inet6
user@router1# set interfaces ms-4/0/0 unit 2 service-domain outside
user@router1# set interfaces xe-0/2/0 unit 0 family inet address 10.0.1.1/30
```

2. Configure IPsec properties such as address, remote-gateway, policies, match-direction, protocol, replay window size, algorithm details, secrecy keys, proposal, authentication method, groups, and version.

```
user@router1# set services ipsec-vpn rule vpn_rule_ms_4_0_01 term term11 from source-address
172.16.0.0/16
user@router1# set services ipsec-vpn rule vpn_rule_ms_4_0_01 term term11 from destination-
address 192.168.0.0/16
user@router1# set services ipsec-vpn rule vpn_rule_ms_4_0_01 term term11 then remote-gateway
10.0.1.2
user@router1# set services ipsec-vpn rule vpn_rule_ms_4_0_01 term term11 then dynamic ike-
policy ike_policy_ms_4_0_0
user@router1# set services ipsec-vpn rule vpn_rule_ms_4_0_01 term term11 then dynamic ipsec-
policy ipsec_policy_ms_4_0_0
user@router1# set services ipsec-vpn rule vpn_rule_ms_4_0_01 term term11 then anti-replay-
window-size 4096
user@router1# set services ipsec-vpn rule vpn_rule_ms_4_0_01 match-direction input
user@router1# set services ipsec-vpn ipsec proposal ipsec_proposal_ms_4_0_0 protocol esp
user@router1# set services ipsec-vpn ipsec proposal ipsec_proposal_ms_4_0_0 authentication-
algorithm hmac-sha1-96
user@router1# set services ipsec-vpn ipsec proposal ipsec_proposal_ms_4_0_0 encryption-
algorithm 3des-cbc
user@router1# set services ipsec-vpn ipsec policy ipsec_policy_ms_4_0_0 perfect-forward-
secrecy keys group2
```

```
user@router1# set services ipsec-vpn ipsec policy ipsec_policy_ms_4_0_0 proposals
ipsec_proposal_ms_4_0_0
user@router1# set services ipsec-vpn ike proposal ike_proposal_ms_4_0_0 authentication-method
pre-shared-keys
user@router1# set services ipsec-vpn ike proposal ike_proposal_ms_4_0_0 dh-group group2
user@router1# set services ipsec-vpn ike policy ike_policy_ms_4_0_0 version 2
user@router1# set services ipsec-vpn ike policy ike_policy_ms_4_0_0 proposals
ike_proposal_ms_4_0_0
user@router1# set services ipsec-vpn ike policy ike_policy_ms_4_0_0 pre-shared-key ascii-text
secret-key
```

3. Configure a service set, the ipsec-vpn options, and rules.

```
user@router1# set services service-set ipsec_ss_ms_4_0_01 next-hop-service inside-service-
interface ms-4/0/0.1
user@router1# set services service-set ipsec_ss_ms_4_0_01 next-hop-service outside-service-
interface ms-4/0/0.2
user@router1# set services service-set ipsec_ss_ms_4_0_01 ipsec-vpn-options local-gateway
10.0.1.1
user@router1# set services service-set ipsec_ss_ms_4_0_01 ipsec-vpn-rules vpn_rule_ms_4_0_01
```

4. Configure routing options static route and next hop.

```
user@router1# set routing-options static route 192.168.0.0/16 next-hop ms-4/0/0.1
```

Results

From the configuration mode of Router 1, confirm your configuration by entering the `show interfaces`, `show services ipsec-vpn`, and `show services service-set` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@router1# show interfaces
ms-4/0/0{
    unit 0 {
        family inet;
    }
    unit 1 {
        family inet;
        family inet6;
        service-domain inside;
    }
    unit 2 {
        family inet;
        family inet6;
        service-domain outside;
    }
}
xe-0/2/0 {
    unit 0 {
        family inet {
            address 10.0.1.1/30;
        }
    }
}
```

```
user@router1# show services ipsec-vpn
rule vpn_rule_ms_4_0_01 {
    term term11 {
        from {
            source-address {
                172.16.0.0/16;
            }
            destination-address {
                192.168.0.0/16;
            }
        }
        then {
            remote-gateway 10.0.1.2;
            dynamic {
                ike-policy ike_policy_ms_4_0_0;
                ipsec-policy ipsec_policy_ms_4_0_0;
            }
            anti-replay-window-size 4096;
        }
    }
    match-direction input;
}
    ipsec {
        proposal ipsec_proposal_ms_4_0_0 {
            protocol esp;
            authentication-algorithm hmac-sha1-96;
            encryption-algorithm 3des-cbc;
        }
        policy ipsec_policy_ms_4_0_0 {
            perfect-forward-secrecy {
                keys group2;
            }
            proposals ipsec_proposal_ms_4_0_0;
        }
    }
    ike {
        proposal ike_proposal_ms_4_0_0 {
            authentication-method pre-shared-keys;
            dh-group group2;
        }
        policy ike_policy_ms_4_0_0 {
            version 2;
            proposals ike_proposal_ms_4_0_0;
            pre-shared-key ascii-text "$9ABC123"; ## SECRET-DATA
        }
    }
```

```
user@router1# show services service-set
ipsec_ss_ms_4_0_01 {
    next-hop-service {
        inside-service-interface ms-4/0/0.1;
        outside-service-interface ms-4/0/0.2;
    }
    ipsec-vpn-options {
        local-gateway 10.0.1.1;
    }
    ipsec-vpn-rules vpn_rule_ms_4_0_01;
}
```

Configuring Router 2

Step-by-Step Procedure

1. Configure the interface properties such as family, service-domain, and unit.

```
user@router2# set interfaces ms-1/0/0 services-options inactivity-non-tcp-timeout 600
user@router2# set interfaces ms-1/0/0 unit 0 family inet
user@router2# set interfaces ms-1/0/0 unit 1 family inet
user@router2# set interfaces ms-1/0/0 unit 1 family inet6
user@router2# set interfaces ms-1/0/0 unit 1 service-domain inside
user@router2# set interfaces ms-1/0/0 unit 2 family inet
user@router2# set interfaces ms-1/0/0 unit 2 family inet6
user@router2# set interfaces ms-1/0/0 unit 2 service-domain outside
user@router2# set interfaces ge-2/0/0 unit 0 family inet adddress 10.0.1.2/30
```

2. Configure IPsec properties such as address, remote-gateway, policies, match-direction, protocol, replay window size, algorithm details, secrecy keys, proposal, authentication method, groups, and version.

```
user@router2# set services ipsec-vpn rule vpn_rule_ms_5_2_01 term term11 from source-address
192.168.0.0/16
user@router2# set services ipsec-vpn rule vpn_rule_ms_5_2_01 term term11 from destination-
address 172.16.0.0/16
user@router2# set services ipsec-vpn rule vpn_rule_ms_5_2_01 term term11 then remote-gateway
10.0.1.1
user@router2# set services ipsec-vpn rule vpn_rule_ms_5_2_01 term term11 then dynamic ike-
policy ike_policy_ms_5_2_0
user@router2# set services ipsec-vpn rule vpn_rule_ms_5_2_01 term term11 then dynamic ipsec-
policy ipsec_policy_ms_5_2_0
user@router2# set services ipsec-vpn rule vpn_rule_ms_5_2_01 term term11 then anti-replay-
window-size 4096
user@router2# set services ipsec-vpn rule vpn_rule_ms_5_2_01 match-direction input
user@router2# set services ipsec-vpn ipsec proposal ipsec_proposal_ms_5_2_0 protocol esp
user@router2# set services ipsec-vpn ipsec proposal ipsec_proposal_ms_5_2_0 authentication-
algorithm hmac-sha1-96
user@router2# set services ipsec-vpn ipsec proposal ipsec_proposal_ms_5_2_0 encryption-
algorithm 3des-cbc
user@router2# set services ipsec-vpn ipsec policy ipsec_policy_ms_5_2_0 perfect-forward-
secrecy keys group2
user@router2# set services ipsec-vpn ipsec policy ipsec_policy_ms_5_2_0 proposals
```

```
ipsec_proposal_ms_5_2_0
user@router2# set services ipsec-vpn ike proposal ike_proposal_ms_5_2_0 authentication-method
pre-shared-keys
user@router2# set services ipsec-vpn ike proposal ike_proposal_ms_5_2_0 dh-group group2
user@router2# set services ipsec-vpn ike policy ike_policy_ms_5_2_0 version 2
user@router2# set services ipsec-vpn ike policy ike_policy_ms_5_2_0 proposals
ike_proposal_ms_5_2_0
user@router2# set services ipsec-vpn ike policy ike_policy_ms_5_2_0 pre-shared-key ascii-text
"$ABC123"
user@router2# set services ipsec-vpn establish-tunnels immediately
```

3. Configure a service set such as next-hop-service, and the ipsec-vpn-options.

```
user@router2# set services service-set ipsec_ss_ms_5_2_01 next-hop-service inside-service-
interface ms-1/0/0.1
user@router2# set services service-set ipsec_ss_ms_5_2_01 next-hop-service outside-service-
interface ms-1/0/0.2
user@router2# set services service-set ipsec_ss_ms_5_2_01 ipsec-vpn-options local-gateway
10.0.1.2
user@router2# set services service-set ipsec_ss_ms_5_2_01 ipsec-vpn-rules vpn_rule_ms_5_2_01
```

4. Configure routing options static route and the next hop.

```
user@router2# set routing-options static route 172.16.0.0/16 next-hop ms-1/0/0.1
```

Results

From the configuration mode of Router 2, confirm your configuration by entering the `show interfaces`, `show services ipsec-vpn`, and `show services service-set` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@router2# show interfaces
ms-1/0/0 {
    unit 0 {
        family inet;
    }
    unit 1 {
        family inet;
        family inet6;
        service-domain inside;
    }
    unit 2 {
        family inet;
        family inet6;
        service-domain outside;
    }
}
ge-2/0/0 {
    unit 0 {
        family inet {
            address 10.0.1.2/30;
        }
    }
}
```

```
user@router2# show services ipsec-vpn
rule vpn_rule_ms_5_2_01 {
    term term11 {
        from {
            source-address {
                192.168.0.0/16;
            }
            destination-address {
                172.16.0.0/16;
            }
        }
        then {
            remote-gateway 10.0.1.1;
            dynamic {
                ike-policy ike_policy_ms_5_2_0;
                ipsec-policy ipsec_policy_ms_5_2_0;
            }
            anti-replay-window-size 4096;
        }
    }
    match-direction input;
}
    ipsec {
        proposal ipsec_proposal_ms_5_2_0 {
            protocol esp;
            authentication-algorithm hmac-sha1-96;
            encryption-algorithm 3des-cbc;
        }
        policy ipsec_policy_ms_5_2_0 {
            perfect-forward-secrecy {
                keys group2;
            }
            proposals ipsec_proposal_ms_5_2_0;
        }
    }
    ike {
        proposal ike_proposal_ms_5_2_0 {
            authentication-method pre-shared-keys;
            dh-group group2;
        }
        policy ike_policy_ms_5_2_0 {
            version 2;
            proposals ike_proposal_ms_5_2_0;
            pre-shared-key ascii-text "$9ABC123"; ## SECRET-DATA
        }
    }
establish-tunnels immediately;
```

```
user@router2# show services service-set
ipsec_ss_ms_5_2_01 {
    next-hop-service {
        inside-service-interface ms-1/0/0.1;
        outside-service-interface ms-1/0/0.2;
    }
    ipsec-vpn-options {
        local-gateway 10.0.1.2;
    }
    ipsec-vpn-rules vpn_rule_ms_5_2_01;
}
```

```
user@router2 #show routing-options
static {
    route 172.16.0.0/16 next-hop ms-1/0/0.1;
}
```

Verification

IN THIS SECTION

- Verifying Tunnel Creation | 791
- Verifying Traffic Flow Through the DEP Tunnel | 792
- Verifying IPsec Security Associations for the Service Set | 793

Verifying Tunnel Creation

Purpose

Verify that Dynamic End Points are created.

Action

Run the following command on Router 1:

```
user@router1 >show services ipsec-vpn ipsec security-associations detail
Service set: ipsec_ss_ms_4_0_01, IKE Routing-instance: default

  Rule: vpn_rule_ms_4_0_01, Term: term11, Tunnel index: 1
  Local gateway: 10.0.1.1, Remote gateway: 10.0.1.2
  IPSec inside interface: ms-4/0/0.1, Tunnel MTU: 1500
  Local identity: ipv4_subnet(any:0,[0..7]=172.16.0.0/16)
  Remote identity: ipv4_subnet(any:0,[0..7]=192.168.0.0/16)

    Direction: inbound, SPI: 112014862, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 24556 seconds
    Hard lifetime: Expires in 25130 seconds
    Anti-replay service: Enabled, Replay window size: 4096

    Direction: outbound, SPI: 1469281276, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 24556 seconds
    Hard lifetime: Expires in 25130 seconds
    Anti-replay service: Enabled, Replay window size: 4096
```

Meaning

The output shows that the IPSec SAs are up on the router with their state as Installed. The IPSec tunnel is up and ready to send traffic over the tunnel.

Verifying Traffic Flow Through the DEP Tunnel

Purpose

Verify traffic flow across the newly-created DEP tunnel.

Action

Run the following command on Router 2:

```
user@router2> show services ipsec-vpn ipsec statistics
PIC: ms-1/0/0, Service set: ipsec_ss_ms_5_2_01

ESP Statistics:
  Encrypted bytes:           153328
  Decrypted bytes:           131424
  Encrypted packets:           2738
  Decrypted packets:           2738
AH Statistics:
  Input bytes:                    0
  Output bytes:                   0
  Input packets:                  0
  Output packets:                 0
Errors:
  AH authentication failures:     0
  ESP authentication failures:    0
  ESP decryption failures:        0
  Bad headers: 0, Bad trailers: 0
  Replay before window drops: 0, Replayed pkts: 0
  IP integrity errors: 0, Exceeds tunnel MTU: 0
  Rule lookup failures: 0, No SA errors: 0
  Flow errors: 0, Misc errors: 0
```

Verifying IPsec Security Associations for the Service Set

Purpose

Verify that the security associations configured for the service set are functioning correctly.

Action

Run the following command on Router 2:

```
user@router2> show services ipsec-vpn ipsec security-associations ipsec_ss_ms_5_2_01
Service set: ipsec_ss_ms_5_2_01, IKE Routing-instance: default

  Rule: vpn_rule_ms_5_2_01, Term: term11, Tunnel index: 1
  Local gateway: 10.0.1.2., Remote gateway: 10.0.1.1
  IPSec inside interface: ms-1/0/0.1, Tunnel MTU: 1500
    Direction SPI         AUX-SPI     Mode       Type     Protocol
    inbound   1612447024  0           tunnel     dynamic  ESP
    outbound  1824720964  0           tunnel     dynamic  ESP
```

Example: Configuring Statically Assigned IPsec Tunnels over a VRF Instance

IN THIS SECTION

- Requirements | 794
- Overview | 794
- Configuration | 794

This example shows how to configure a statically assigned IPsec tunnel over a VRF instance, and contains the following sections:

Requirements

This example uses the following hardware and software components:

- M Series, MX Series, or T Series router that is configured as a provider edge router.
- Junos OS Release 9.4 and later.

No special configuration beyond device initialization is required before you can configure this feature.

Overview

Junos OS enables you to configure statically assigned IPsec tunnels on Virtual Routing and Forwarding (VRF) instances. Ability to configure IPsec tunnels on VRF instances enhances network segmentation and security. You can have multiple customer tunnels configured on the same PE router over VRF instances. Each VRF instance acts as logical router with an exclusive routing table.

Configuration

IN THIS SECTION

- Configuring the Provider Edge Router | 794
- Results | 798

This example shows the configuration of an IPsec tunnel over a VRF instance on a provider edge router, and provides step-by-step instructions for completing the required configuration.

This section contains:

Configuring the Provider Edge Router

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the [edit] hierarchy level.

```
set interfaces ge-0/3/0 unit 0 family inet address 10.6.6.6/32
set interfaces ge-1/1/0 description "teller ge-0/1/0"
set interfaces ge-1/1/0 unit 0 family inet address 10.21.1.1/16
set interfaces ms-1/2/0 unit 0 family inet address 10.7.7.7/32
set interfaces ms-1/2/0 unit 1 family inet
set interfaces ms-1/2/0 unit 1 service-domain inside
set interfaces ms-1/2/0 unit 2 family inet
set interfaces ms-1/2/0 unit 2 service-domain outside
set policy-options policy-statement vpn-export then community add vpn-community
set policy-options policy-statement vpn-export then accept
set policy-options policy-statement vpn-import term a from community vpn-community
set policy-options policy-statement vpn-import term a then accept
set policy-options community vpn-community members target:100:20
set routing-instances vrf instance-type vrf
set routing-instances vrf interface ge-0/3/0.0
set routing-instances vrf interface ms-1/2/0.1
set routing-instances vrf route-distinguisher 192.168.0.1:1
set routing-instances vrf vrf-import vpn-import
set routing-instances vrf vrf-export vpn-export
set routing-instances vrf routing-options static route 10.0.0.0/0 next-hop ge-0/3/0.0
set routing-instances vrf routing-options static route 10.11.11.1/32 next-hop ge-0/3/0.0
set routing-instances vrf routing-options static route 10.8.8.1/32 next-hop ms-1/2/0.1
set services ipsec-vpn ipsec proposal demo_ipsec_proposal protocol esp
set services ipsec-vpn ipsec proposal demo_ipsec_proposal authentication-algorithm hmac-sha1-96
set services ipsec-vpn ipsec proposal demo_ipsec_proposal encryption-algorithm 3des-cbc
set services ipsec-vpn ipsec policy demo_ipsec_policy perfect-forward-secrecy keys group2
set services ipsec-vpn ipsec policy demo_ipsec_policy proposals demo_ipsec_proposal
set services ipsec-vpn ike proposal demo_ike_proposal authentication-method pre-shared-keys
set services ipsec-vpn ike proposal demo_ike_proposal dh-group group2
set services ipsec-vpn ike policy demo_ike_policy proposals demo_ike_proposal
set services ipsec-vpn ike policy demo_ike_policy pre-shared-key ascii-text juniperkey
set services ipsec-vpn rule demo-rule term demo-term then remote-gateway 10.21.2.1
set services ipsec-vpn rule demo-rule term demo-term then dynamic ike-policy demo_ike_policy
set services ipsec-vpn rule demo-rule match-direction input
set services service-set demo-service-set next-hop-service inside-service-interface ms-1/2/0.1
set services service-set demo-service-set next-hop-service outside-service-interface ms-1/2/0.2
set services service-set demo-service-set ipsec-vpn-options local-gateway 10.21.1.1
set services service-set demo-service-set ipsec-vpn-rules demo-rule
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure a statically assigned IPsec tunnel on a VRF instance:

1. Configure the interfaces. In this step, you configure two Ethernet (ge) interfaces, one services interface (ms-), and also the service-domain properties for the logical interfaces of the services interface. Note that the logical interface that is marked as the inside interface applies the configured service on the traffic, whereas the one that is marked as the outside interface acts as the egress point for the traffic on which the inside interface has applied the service.

```
[edit interfaces]
user@PE1# set ge-0/3/0 unit 0 family inet address 10.6.6.6/32
user@PE1# set ge-1/1/0 description "teller ge-0/1/0"
user@PE1# set ge-1/1/0 unit 0 family inet address 10.21.1.1/16
user@PE1# set ms-1/2/0 unit 0 family inet address 10.7.7.7/32
user@PE1# set ms-1/2/0 unit 1 family inet
user@PE1# set ms-1/2/0 unit 1 service-domain inside
user@PE1# set ms-1/2/0 unit 2 family inet
user@PE1# set ms-1/2/0 unit 2 service-domain outside
```

2. Configure a routing policy to specify route import and export criteria for the VRF instance. The import and export policies defined in this step are referenced from the routing-instance configuration in the next step.

```
[edit policy-options]
user@PE1# set policy-statement vpn-export then community add vpn-community
user@PE1# set policy-statement vpn-export then accept
user@PE1# set policy-statement vpn-import term a from community vpn-community
user@PE1# set policy-statement vpn-import term a then accept
user@PE1# set community vpn-community members target:100:20
```

3. Configure a routing instance and specify the routing-instance type as vrf. Apply the import and export policies defined in the previous step to the routing instance, and specify a static route to send the IPsec traffic to the inside interface (ms-1/2/0.1) configured in the first step.

```
[edit routing-instance]
user@PE1# set vrf instance-type vrf
user@PE1# set vrf interface ge-0/3/0.0
user@PE1# set vrf interface ms-1/2/0.1
user@PE1# set vrf route-distinguisher 192.168.0.1:1
user@PE1# set vrf vrf-import vpn-import
user@PE1# set vrf vrf-export vpn-export
```

```
user@PE1# set vrf routing-options static route 10.0.0.0/0 next-hop ge-0/3/0.0
user@PE1# set vrf routing-options static route 10.11.11.1/32 next-hop ge-0/3/0.0
user@PE1# set vrf routing-options static route 10.8.8.1/32 next-hop ms-1/2/0.1
```

4. Configure IKE and IPsec proposals and policies, and a rule to apply the IKE policy on the incoming traffic..

> **NOTE**: By default, Junos OS uses IKE policy version 1.0. Junos OS Release 11.4 and later also support IKE policy version 2.0 which you must configure at `[edit services ipsec-vpn ike policy policy-name pre-shared]`.

```
[edit services]
user@PE1# set ipsec-vpn ipsec proposal demo_ipsec_proposal protocol esp
user@PE1# set ipsec-vpn ipsec proposal demo_ipsec_proposal authentication-algorithm hmac-
sha1-96
user@PE1# set ipsec-vpn ipsec proposal demo_ipsec_proposal encryption-algorithm 3des-cbc
user@PE1# set ipsec-vpn ipsec policy demo_ipsec_policy perfect-forward-secrecy keys group2
user@PE1# set ipsec-vpn ipsec policy demo_ipsec_policy proposals demo_ipsec_proposal
user@PE1# set ipsec-vpn ike proposal demo_ike_proposal authentication-method pre-shared-keys
user@PE1# set ipsec-vpn ike proposal demo_ike_proposal dh-group group2
user@PE1# set ipsec-vpn ike policy demo_ike_policy proposals demo_ike_proposal
user@PE1# set ipsec-vpn ike policy demo_ike_policy pre-shared-key ascii-text juniperkey
user@PE1# set ipsec-vpn rule demo-rule term demo-term then remote-gateway 10.21.2.1
user@PE1# set ipsec-vpn rule demo-rule term demo-term then dynamic ike-policy demo_ike_policy
user@PE1# set ipsec-vpn rule demo-rule match-direction input
```

5. Configure a next-hop style service set. Note that you must configure the inside and outside interfaces that you configured in the first step as the `inside-service-interface` and `outside-service-interface` respectively.

```
[edit services]
user@PE1# set service-set demo-service-set next-hop-service inside-service-interface
ms-1/2/0.1
user@PE1# set service-set demo-service-set next-hop-service outside-service-interface
ms-1/2/0.2
user@PE1# set service-set demo-service-set ipsec-vpn-options local-gateway 10.21.1.1
user@PE1# set service-set demo-service-set ipsec-vpn-rules demo-rule
```

6. Commit the configuration.

```
[edit]
user@PE1# commit
```

Results

From the configuration mode of Router 1, confirm your configuration by entering the `show interfaces`, `show policy-options`, `show routing-instances`, `show services ipsec-vpn`, and `show services service-set` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@PE1# show interfaces
...
    ms-1/2/0 {
        unit 0 {
            family inet {
                address 10.7.7.7/32;
            }
        }
        unit 1 {
            family inet;
            service-domain inside;
        }
        unit 2 {
            family inet;
            service-domain outside;
        }
    }
    ge-0/3/0 {
        unit 0 {
            family inet {
                address 10.6.6.6/32;
            }
        }
    }
    ge-1/1/0 {
        description "teller ge-0/1/0";
        unit 0 {
            family inet {
                address 10.21.1.1/16;
            }
        }
    }
...
```

```
user@PE1# show policy-options
    policy-statement vpn-export {
        then {
            community add vpn-community;
            accept;
        }
    }
    policy-statement vpn-import {
        term a {
            from community vpn-community;
            then accept;
        }
    }
    community vpn-community members target:100:20;
```

```
user@PE1# show routing-instances
vrf {
    instance-type vrf;
    interface ge-0/3/0.0;
    interface ms-1/2/0.1;
    route-distinguisher 192.168.0.1:1;
    vrf-import vpn-import;
    vrf-export vpn-export;
    routing-options {
        static {
            route 10.0.0.0/0 next-hop ge-0/3/0.0;
            route 10.11.11.1/32 next-hop ge-0/3/0.0;
            route 10.8.8.1/32 next-hop ms-1/2/0.1;
        }
    }
}
```

```
user@PE1# show services ipsec-vpn
ipsec-vpn {
    rule demo-rule {
        term demo-term {
            then {
                remote-gateway 10.21.2.1;
                dynamic {
                    ike-policy demo_ike_policy;
                }
            }
        }
        match-direction input;
    }
    ipsec {
        proposal demo_ipsec_proposal {
            protocol esp;
            authentication-algorithm hmac-sha1-96;
            encryption-algorithm 3des-cbc;
        }
        policy demo_ipsec_policy {
            perfect-forward-secrecy {
                keys group2;
            }
            proposals demo_ipsec_proposal;
        }
    }
    ike {
        proposal demo_ike_proposal {
            authentication-method pre-shared-keys;
            dh-group group2;
        }
        policy demo_ike_policy {
            proposals demo_ike_proposal;
            pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
        }
    }
}
```

```
user@PE1# show services service-set demo-service-set
    next-hop-service {
        inside-service-interface ms-1/2/0.1;
        outside-service-interface ms-1/2/0.2;
    }
    ipsec-vpn-options {
        local-gateway 10.21.1.1;
    }
    ipsec-vpn-rules demo-rule;
```

SEE ALSO

- Understanding Junos VPN Site Secure | 623
- Configuring Security Associations | 645
- Configuring IPsec Proposals | 688

Multitask Example: Configuring IPsec Services

IN THIS SECTION

- Configuring the IKE Proposal | 801
- Configuring the IKE Policy (and Referencing the IKE Proposal) | 803
- Configuring the IPsec Proposal | 804
- Configuring the IPsec Policy (and Referencing the IPsec Proposal) | 805
- Configuring the IPsec Rule (and Referencing the IKE and IPsec Policies) | 806
- Configuring IPsec Trace Options | 807
- Configuring the Access Profile (and Referencing the IKE and IPsec Policies) | 808
- Configuring the Service Set (and Referencing the IKE Profile and the IPsec Rule) | 810

The following example-based instructions show how to configure IPsec services. The configuration involves defining an IKE policy, an IPsec policy, IPsec rules, trace options, and service sets.

This topic includes the following tasks:

Configuring the IKE Proposal

The IKE proposal configuration defines the algorithms and keys used to establish the secure IKE connection with the peer security gateway. For more information about IKE proposals, see "Configuring IKE Proposals" on page 675 .

To define the IKE proposal:

1. In configuration mode, go to the following hierarchy level:

```
user@host# edit services ipsec-vpn
```

2. Configure the authentication method, which is `pre-shared keys` in this example:

```
[edit services ipsec-vpn]
user@host# set ike proposal test-IKE-proposal authentication-method pre-shared-keys
```

3. Configure the Diffie-Hellman Group and specify a name—for example, `group1`:

```
[edit services ipsec-vpn]
user@host# set ike proposal test-IKE-proposal dh-group group1
```

4. Configure the authentication algorithm, which is `sha1` in this example:

```
[edit services ipsec-vpn]
user@host# set ike proposal test-IKE-proposal authentication-algorithm sha1
```

5. Configure the encryption algorithm, which is `aes-256-cbc` in this example:

```
[edit services ipsec-vpn]
user@host# set ike proposal test-IKE-proposal encryption-algorithm aes-256-cbc
```

The following sample output shows the configuration of the IKE proposal:

```
[edit services ipsec-vpn]
user@host# show ike
proposal test-IKE-proposal {
    authentication-method pre-shared-keys;
    dh-group group1;
    authentication-algorithm sha1;
    encryption-algorithm aes-256-cbc;
}
```

SEE ALSO

Configuring IKE Proposals | 675

Configuring the IKE Policy (and Referencing the IKE Proposal)

The IKE policy configuration defines the proposal, mode, addresses, and other security parameters used during IKE negotiation. For more information about IKE policies, see "Configuring IKE Policies" on page 681 .

To define the IKE policy and reference the IKE proposal:

1. In configuration mode, go to the following hierarchy level:

   ```
   user@host# edit services ipsec-vpn
   ```

2. Configure the IKE first phase mode—for example, `main`:

   ```
   [edit services ipsec-vpn]
   user@host# set ike policy test-IKE-policy mode main
   ```

3. Configure the proposal, which is `test-IKE-proposal` in this example:

   ```
   [edit services ipsec-vpn]
   user@host# set ike policy test-IKE-policy proposals test-IKE-proposal
   ```

4. Configure the local identification with an IPv4 address—for example, `192.168.255.2`:

   ```
   [edit services ipsec-vpn]
   user@host# set ike policy test-IKE-policy local-id ipv4_addr 192.168.255.2
   ```

5. Configure the preshared key in ASCII text format, which is `TEST` in this example:

   ```
   [edit services ipsec-vpn]
   user@host# set ike policy test-IKE-policy pre-shared-key ascii-text TEST
   ```

The following sample output shows the configuration of the IKE policy:

```
[edit services ipsec-vpn]
user@host# show ike
```

```
policy test-IKE-policy {
    mode main;
    proposals test-IKE-proposal;
    local-id ipv4_addr 192.168.255.2;
    pre-shared-key ascii-text TEST;
}
```

Configuring the IPsec Proposal

The IPsec proposal configuration defines the protocols and algorithms (security services) that are required to negotiate with the remote IPsec peer. For more information about IPsec proposals, see "Configuring IPsec Proposals" on page 688 .

To define the IPsec proposal:

1. In configuration mode, go to the following hierarchy level:

   ```
   user@host# edit services ipsec-vpn
   ```

2. Configure the IPsec protocol for the proposal—for example, `esp`:

   ```
   [edit services ipsec-vpn]
   user@host# set ipsec proposal test-IPsec-proposal protocol esp
   ```

3. Configure the authentication algorithm for the proposal, which is `hmac-sha1-96` in this example:

   ```
   [edit services ipsec-vpn]
   user@host# set ipsec proposal test-IPsec-proposal authentication-algorithm hmac-sha1-96
   ```

4. Configure the encryption algorithm for the proposal, which is `aes-256-cbc` in this example:

   ```
   [edit services ipsec-vpn]
   user@host# set ipsec proposal test-IPsec-proposal encryption-algorithm aes-256-cbc
   ```

The following sample output shows the configuration of the IPsec proposal:

```
[edit services ipsec-vpn]
user@host# show ike
proposal test-IPsec-proposal {
    protocol esp;
```

```
    authentication-algorithm hmac-sha1-96;
    encryption-algorithm aes-256-cbc;
}
```

SEE ALSO

Configuring IPsec Proposals | 688

Configuring the IPsec Policy (and Referencing the IPsec Proposal)

The IPsec policy configuration defines a combination of security parameters (IPsec proposals) used during IPsec negotiation. It defines PFS and the proposals needed for the connection. For more information about IPsec policies, see "Configuring IPsec Policies" on page 694 .

To define the IPsec policy and reference the IPsec proposal:

1. In configuration mode, go to the following hierarchy level:

   ```
   user@host# edit services ipsec-vpn
   ```

2. Configure the keys for perfect forward secrecy in the IPsec policy—for example, `group1`:

   ```
   [edit services ipsec-vpn]
   user@host# set ipsec policy test-IPsec-policy perfect-forward-secrecy keys group1
   ```

3. Configure a set of IPsec proposals in the IPsec policy—for example, `test-IPsec-proposal`:

   ```
   [edit services ipsec-vpn]
   user@host# set ipsec policy test-IPsec-policy proposals test-IPsec-proposal
   ```

The following sample output shows the configuration of the IPsec policy:

```
[edit services ipsec-vpn]
user@host# show ipsec policy test-IPsec-policy
perfect-forward-secrecy {
    keys group1;
}
proposals test-IPsec-proposal;
```

SEE ALSO

Configuring IPsec Policies | 694

Configuring the IPsec Rule (and Referencing the IKE and IPsec Policies)

The IPsec rule configuration defines the direction that specifies whether the match is applied on the input or output side of the interface. The configuration also consists of a set of terms that specify the match conditions and applications that are included and excluded and also specify the actions and action modifiers to be performed by the router software. For more information about IPsec rules, see "Configuring IPsec Rules" on page 723 .

To define the IPsec rule and reference the IKE and IPsec policies:

1. In configuration mode, go to the following hierarchy level:

```
user@host# edit services ipsec-vpn
```

2. Configure the IP destination address for the IPsec term in the IPsec rule—for example, `192.168.255.2/32`:

```
[edit services ipsec-vpn]
user@host# set rule test-IPsec-rule term 10 from destination-address 192.168.255.2/32
```

3. Configure the remote gateway address for the IPsec term in the IPsec rule—for example, `0.0.0.0`:

```
[edit services ipsec-vpn]
user@host# set rule test-IPsec-rule term 10 then remote-gateway 0.0.0.0
```

4. Configure a dynamic security association for IKE policy for the IPsec term in the IPsec rule, which is `test-IKE-policy` in this example:

```
[edit services ipsec-vpn]
user@host# set rule test-IPsec-rule term 10 then dynamic ike-policy test-IKE-policy
```

5. Configure a dynamic security association for IKE proposal for the IPsec term in the IPsec rule, which is `test-IPsec-proposal` in this example:

```
[edit services ipsec-vpn]
user@host# set rule test-IPsec-rule term 10 then dynamic ipsec-policy test-IPsec-policy
```

6. Configure a direction for which the rule match is being applied in the IPsec rule—for example, `input`:

   ```
   [edit services ipsec-vpn]
   user@host# set rule test-IPsec-rule match-direction input
   ```

The following sample output shows the configuration of the IPsec rule:

```
[edit services ipsec-vpn]
user@host# show rule test-IPsec-rule
term 10 {
    from {
        destination-address {
            192.168.255.2/32;
        }
    }
    then {
        remote-gateway 0.0.0.0;
        dynamic {
            ike-policy test-IKE-policy;
            ipsec-policy test-IPsec-policy;
        }
    }
}
match-direction input;
```

Configuring IPsec Trace Options

The IPsec trace options configuration tracks IPsec events and records them in a log file in the **/var/log** directory. By default, this file is named **/var/log/kmd**. For more information about IPsec rules, see "Tracing Junos VPN Site Secure Operations" on page 812 .

To define the IPsec trace options:

1. In configuration mode, go to the following hierarchy level:

   ```
   user@host# edit services ipsec-vpn
   ```

2. Configure the trace file, which is `ipsec.log` in this example:

```
[edit services ipsec-vpn]
user@host# set traceoptions file ipsec.log
```

3. Configure all the tracing parameters with the option `all` in this example:

```
[edit services ipsec-vpn]
user@host# set traceoptions flag all
```

The following sample output shows the configuration of the IPsec trace options:

```
[edit services ipsec-vpn]
user@host# show traceoptions
file ipsec.log;
flag all;
```

Configuring the Access Profile (and Referencing the IKE and IPsec Policies)

The access profile configuration defines the access profile and references the IKE and IPsec policies. For more information about access profile, see *Configuring an IKE Access Profile*.

To define the access profile and reference the IKE and IPsec policies:

1. In configuration mode, go to the following hierarchy level:

```
user@host# [edit access]
```

2. Configure the list of local and remote proxy identity pairs with the `allowed-proxy-pair` option. In this example, `10.0.0.0/24` is the IP address for local proxy identity and `10.0.1.0/24` is the IP address for remote proxy identity:

```
[edit access]
user@host# set profile IKE-profile-TEST client * ike allowed-proxy-pair local 10.0.0.0/24
remote 10.0.1.0/24
```

3. Configure the IKE policy—for example, `test-IKE-policy`:

```
[edit access]
user@host# set profile IKE-profile-TEST client * ike ike-policy test-IKE-policy
```

4. Configure the IPsec policy—for example, `test-IPsec-policy`:

```
[edit access]
user@host# set profile IKE-profile-TEST client * ike ipsec-policy test-IPsec-policy
```

5. Configure the identity of logical service interface pool, which is `TEST-intf` in this example:

```
[edit access]
user@host# set profile IKE-profile-TEST client * ike interface-id TEST-intf
```

The following sample output shows the configuration of the access profile:

```
[edit access]
user@host# show
profile IKE-profile-TEST {
    client * {
        ike {
            allowed-proxy-pair local 10.0.0.0/24 remote 10.0.1.0/24;
            ike-policy test-IKE-policy;
            ipsec-policy test-IPsec-policy; # new statement
            interface-id TEST-intf;
        }
    }
}
```

SEE ALSO

Configuring an IKE Access Profile

Configuring the Service Set (and Referencing the IKE Profile and the IPsec Rule)

The service set configuration defines IPsec service sets that require additional specifications and references the IKE profile and the IPsec rule. For more information about IPsec service sets, see "Configuring IPsec Service Sets" on page 734 .

To define the service set configuration with the next-hop service sets and IPsec VPN options:

1. In configuration mode, go to the following hierarchy level:

```
user@host# [edit services]
```

2. Configure a service set with parameters for next hop service interfaces for the inside network—for example, `sp-1/2/0.1`:

```
[edit services]
user@host# set service-set TEST next-hop-service inside-service-interface sp-1/2/0.1
```

3. Configure a service set with parameters for next hop service interfaces for the outside network—for example, `sp-1/2/0.2`:

```
[edit services]
user@host# set service-set TEST next-hop-service outside-service-interface sp-1/2/0.2
```

4. Configure the IPsec VPN options with the address and routing instance for the local gateway—for example, `192.168.255.2`:

```
[edit services]
user@host# set service-set TEST ipsec-vpn-options local-gateway 192.168.255.2
```

5. Configure the IPsec VPN options with the IKE access profile for dynamic peers, which is `IKE-profile-TEST` in this example:

```
[edit services]
user@host# set service-set TEST ipsec-vpn-options ike-access-profile IKE-profile-TEST
```

6. Configure a service set with IPsec VPN rules, which is `test-IPsec-rule` in this example:

```
[edit services]
user@host# set service-set TEST ipsec-vpn-rules test-IPsec-rule
```

The following sample output shows the configuration of the service set configuration referencing the IKE profile and the IPsec rule:

```
[edit services]user@host# show service-set TEST
next-hop-service {
    inside-service-interface sp-1/2/0.1;
    outside-service-interface sp-1/2/0.2;
}
ipsec-vpn-options {
    local-gateway 192.168.255.2;
    ike-access-profile IKE-profile-TEST;
}
ipsec-vpn-rules test-IPsec-rule;
```

SEE ALSO

Configuring IPsec Service Sets | 734

Disabling NAT-T on MX Series Routers for Handling NAT with IPsec-Protected Packets

Before Junos OS Release 17.4R1, Network Address Translation-Traversal (NAT-T) is not supported for the Junos VPN Site Secure suite of IPsec features on the MX Series routers. By default, Junos OS detects whether either one of the IPsec tunnels is behind a NAT device and automatically switches to using NAT-T for the protected traffic. To avoid running unsupported NAT-T in Junos OS releases before 17.4R1, you must disable NAT-T by including the `disable-natt` statement at the `[edit services ipsec-vpn]` hierarchy level. When you disable NAT-T, the NAT-T functionality is globally switched off. When you disable NAT-T and a NAT device is present between the two IPsec gateways, ISAKMP messages are negotiated using UDP port 500 and data packets are encapsulated with Encapsulating Security Payload (ESP).

Network Address Translation-Traversal (NAT-T) is a method for getting around IP address translation issues encountered when data protected by IPsec passes through a NAT device for address translation. Any changes to the IP addressing, which is the function of NAT, causes IKE to discard packets. After detecting one or more NAT devices along the data path during Phase 1 exchanges, NAT-T adds a layer of User Datagram Protocol (UDP) encapsulation to IPsec packets so they are not discarded after address translation. NAT-T encapsulates both IKE and ESP traffic within UDP with port 4500 used as both the source and destination port. Because NAT devices age out stale UDP translations, keepalive messages are required between the peers.

The location of a NAT device can be such that:

- Only the IKEv1 or IKEv2 initiator is behind a NAT device. Multiple initiators can be behind separate NAT devices. Initiators can also connect to the responder through multiple NAT devices.
- Only the IKEv1 or IKEv2 responder is behind a NAT device.
- Both the IKEv1 or IKEv2 initiator and the responder are behind a NAT device.

Dynamic endpoint VPN covers the situation where the initiator's IKE external address is not fixed and is therefore not known by the responder. This can occur when the initiator's address is dynamically assigned by an ISP or when the initiator's connection crosses a dynamic NAT device that allocates addresses from a dynamic address pool.

Configuration examples for NAT-T are provided for the topology in which only the responder is behind a NAT device and the topology in which both the initiator and responder are behind a NAT device. Site-to-site IKE gateway configuration for NAT-T is supported on both the initiator and responder. A remote IKE ID is used to validate a peer’s local IKE ID during Phase 1 of IKE tunnel negotiation. Both the initiator and responder require a local identify and remote identity string.

SEE ALSO

> *disable-natt*

Tracing Junos VPN Site Secure Operations

IN THIS SECTION

- Disabling IPsec Tunnel Endpoint in Traceroute | **814**
- Tracing IPsec PKI Operations | **814**

NOTE: Junos VPN Site Secure is a suite of IPsec features supported on multiservices line cards (MS-DPC, MS-MPC, and MS-MIC), and was previously referred to as IPsec services.

Trace operations track IPsec events and record them in a log file in the /var/log directory. By default, this file is named /var/log/kmd.

To trace IPsec operations, include the traceoptions statement at the [edit services ipsec-vpn] hierarchy level:

```
[edit services ipsec-vpn]
traceoptions {
    file <filename> <files number> <match regular-expression> <size bytes> <world-readable | no-
world-readable>;
    flag flag;
    level level;
    no-remote-trace;
}
```

You can specify the following IPsec tracing flags:

- all—Trace everything.
- certificates—Trace certificates events.
- database—Trace security associations database events.
- general—Trace general events.
- ike—Trace IKE module processing.
- parse—Trace configuration processing.
- policy-manager—Trace policy manager processing.
- routing-socket—Trace routing socket messages.
- snmp—Trace SNMP operations.
- timer—Trace internal timer events.

The level statement sets the key management process (kmd) tracing level. The following values are supported:

- all—Match all levels.

- error—Match error conditions.
- info–Match informational messages.
- notice—Match conditions that should be handled specially.
- verbose—Match verbose messages.
- warning—Match warning messages.

This section includes the following topics:

Disabling IPsec Tunnel Endpoint in Traceroute

If you include the no-ipsec-tunnel-in-traceroute statement at the [edit services ipsec-vpn] hierarchy level, the IPsec tunnel is not treated as a next hop and the time to live (TTL) is not decremented. Also, if the TTL reaches zero, an ICMP time exceeded message is not generated.

```
[edit services ipsec-vpn]
no-ipsec-tunnel-in-traceroute;
```

NOTE: This functionality is also provided by the passive-mode-tunneling statement. You can use the no-ipsec-tunnel-in-traceroute statement in specific scenarios in which the IPsec tunnel should not be treated as a next hop and passive mode is not desired.

Tracing IPsec PKI Operations

Trace operations track IPsec PKI events and record them in a log file in the /var/log directory. By default, this file is named /var/log/pkid.

To trace IPsec PKI operations, include the traceoptions statement at the [edit security pki] hierarchy level:

```
[edit security pki]
traceoptions {
    file filename <files number> <match regular-expression> <size maximum-file-size> <world-
readable | no-world-readable>;
    flag flag (all | certificate-verification | enrollment | online-crl-check);
}
```

You can specify the following PKI tracing flags:

- `all`—Trace everything.
- `certificates`—Trace certificates events.
- `database`—Trace security associations database events.
- `general`—Trace general events.
- `ike`—Trace IKE module processing.
- `parse`—Trace configuration processing.
- `policy-manager`—Trace policy manager processing.
- `routing-socket`—Trace routing socket messages.
- `snmp`—Trace SNMP operations.
- `timer`—Trace internal timer events.

RELATED DOCUMENTATION

Configuring IKE Policies | 681

Configuring IKE Proposals | 675

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
18.2R1	Starting in Junos OS Release 18.2R1, you can configure the MX Series router with MS-MPCs or MS-MICs to send only the end-entity certificate for certificate-based IKE authentication instead of the full certificate chain.
17.2R1	Starting in Junos OS Release 17.2R1, tou can use the `gw-interface` statement to enable the cleanup of IKE triggers and IKE and IPsec SAs when an IPsec tunnel's local gateway IP address goes down, or the MS-MIC or MS-MPC being used in the tunnel's service set goes down.
17.1	Starting in Junos OS Release 17.1, AMS supports IPSec tunnel distribution
16.1	Starting in Junos OS Release 16.1, to configure link-type tunnels, (i.e., next-hop style), for HA purposes, you can configure AMS logical interfaces as the IPsec internal interfaces by using the `ipsec-inside-interface` *`interface-name`* statement at the `[edit services ipsec-vpn rule` *`rule-name`* `term` *`term-name`* `from]` hierarchy level.

16.1	Starting in Junos OS Release 16.1, you can enable multipath forwarding of IPsec traffic by configuring UDP encapsulation in the service set, which adds a UDP header to the IPsec encapsulation of packets.
14.2	Starting in Junos OS Release 14.2, passive mode tunneling is supported on MS-MICs and MS-MPCs.
14.2	Starting in Junos OS Release 14.2, the `header-integrity-check` option that is supported on MS-MICs and MS-MPCs to verify the packet header for anomalies in IP, TCP, UDP, and ICMP information and flag such anomalies and errors has a functionality that is opposite to the functionality caused by passive mode tunneling.
14.1	Starting in Junos OS Release 14.1, in packets that are transmitted through dynamic endpoint IPSec tunnels, you can enable the value set in the DF bit of the packet entering the tunnel to be copied only to the outer header of the IPsec packet and to not cause any modification to the DF bit in the inner header of the IPsec packet.

CHAPTER 31

IPsec Tunnels With Dynamic Endpoints

IN THIS CHAPTER

- IPsec Tunnels With Dynamic Endpoints | 817

IPsec Tunnels With Dynamic Endpoints

IN THIS SECTION

- Configuring Dynamic Endpoints for IPsec Tunnels | 817
- Example: Configuring Dynamically Assigned Policy Based Tunnels | 825

Configuring Dynamic Endpoints for IPsec Tunnels

IN THIS SECTION

- Authentication Process | 818
- Implicit Dynamic Rules | 818
- Reverse Route Insertion | 819
- Configuring an IKE Access Profile | 820
- Referencing the IKE Access Profile in a Service Set | 821
- Configuring the Interface Identifier | 822
- Default IKE and IPsec Proposals | 822
- Distributing Endpoint IPsec Tunnels Among Services Interfaces | 823

IPsec tunnels can also be established using *dynamic peer* security gateways, in which the remote ends of tunnels do not have a statically assigned IP address. Since the remote address is not known and might be pulled from an address pool each time the remote host reboots, establishment of the tunnel relies on using IKE `main` mode with either preshared global keys or digital certificates that accept any remote identification value. Both policy-based and link-type tunnels are supported:

- Policy-based tunnels used shared mode.
- Link-type or routed tunnels use dedicated mode. Each tunnel allocates a services interface from a pool of interfaces configured for the dynamic peers. Routing protocols can be configured to run on these services interfaces to learn routes over the IPsec tunnel that is used as a link in this scenario.

This section includes the following topics:

Authentication Process

The remote (dynamic peer) initiates the negotiations with the local (Juniper Networks) router. The local router uses the default IKE and IPsec policies to match the proposals sent by the remote peer to negotiate the security association (SA) values. Implicit proposals contain a list of all the supported transforms that the local router expects from all the dynamic peers.

If preshared key authentication is used, the preshared key is global for a service set. When seeking the preshared key for the peer, the local router matches the peer's source address against any explicitly configured preshared keys in that service set. If a match is not found, the local router uses the global preshared key for authentication.

Phase 2 of the authentication matches the *proxy identities* of the protected hosts and networks sent by the peer against a list of configured proxy identities. The accepted proxy identity is used to create the dynamic rules for encrypting the traffic. You can configure proxy identities by including the `allowed-proxy-pair` statement in the IKE access profile. If no entry matches, the negotiation is rejected.

If you do not configure the `allowed-proxy-pair` statement, the default value `ANY(0.0.0.0/0)-ANY` is applied, and the local router accepts any proxy identities sent by the peer. Both IPv4 and IPv6 addresses are accepted, but you must configure all IPv6 addresses manually.

Once the phase 2 negotiation completes successfully, the router builds the dynamic rules and inserts the reverse route into the routing table using the accepted proxy identity.

Implicit Dynamic Rules

After successful negotiation with the dynamic peer, the key management process (kmd) creates a dynamic rule for the accepted phase 2 proxy and applies it on the local AS or Multiservices PIC. The source and destination addresses are specified by the accepted proxy. This rule is used to encrypt traffic directed to one of the end hosts in the phase 2 proxy identity.

The dynamic rule includes an `ipsec-inside-interface` value, which is the interface name assigned to the dynamic tunnel. The `source-address` and `destination-address` values are accepted from the proxy ID. The `match-direction` value is `input` for next-hop-style service sets.

> **NOTE**: You do not configure this rule; it is created by the key management process (kmd).

Rule lookup for static tunnels is unaffected by the presence of a dynamic rule; it is performed in the order configured. When a packet is received for a service set, static rules are always matched first.

Dynamic rules are matched after the rule match for static rules has failed.

Response to dead peer detection (DPD) hello messages takes place the same way with dynamic peers as with static peers. Initiating DPD hello messages from dynamic peers is not supported.

Reverse Route Insertion

Static routes are automatically inserted into the route table for those networks and hosts protected by a remote tunnel endpoint. These protected hosts and networks are known as remote proxy identities.

Each route is created based on the remote proxy network and mask sent by the peer and is inserted in the relevant route table after successful phase 1 and phase 2 negotiations.

The route preference for each static reverse route is 1. This value is necessary to avoid conflict with similar routes that might be added by the routing protocol process (rpd).

No routes are added if the accepted remote proxy address is the default (`0.0.0.0/0`). In this case you can run routing protocols over the IPsec tunnel to learn routes and add static routes for the traffic you want to be protected over this tunnel.

For next-hop-style service sets, the reverse routes include next hops pointing to the locations specified by the `inside-service-interface` statement.

The route table in which to insert these routes depends on where the `inside-service-interface` location is listed. If these interfaces are present in a VPN routing and forwarding (VRF) instance, then routes are added to the corresponding VRF table; otherwise, the routes are added to `inet.0`.

> **NOTE**: Reverse route insertion takes place only for tunnels to dynamic peers. These routes are added only for next-hop-style service sets.

Configuring an IKE Access Profile

You can configure only one tunnel profile per service set for all dynamic peers. The configured preshared key in the profile is used for IKE authentication of all dynamic peers terminating in that service set. Alternatively, you can include the `ike-policy` statement to reference an IKE policy you define with either specific identification values or a wildcard (the `any-remote-id` option). You configure the IKE policy at the `[edit services ipsec-vpn ike]` hierarchy level.

The IKE tunnel profile specifies all the information needed to complete the IKE negotiation. Each protocol has its own statement hierarchy within the client statement to configure protocol-specific attribute value pairs, but only one client configuration is allowed for each profile. The following is the configuration at the `[edit access]` hierarchy level; for more information on access profiles, see the Junos OS Administration Library for Routing Devices.

```
[edit access]
profile profile-name {
    client * {
        ike {
            allowed-proxy-pair {
                remote remote-proxy-address local local-proxy-address;
            }
            pre-shared-key (ascii-text key-string | hexadecimal key-string);
            ike-policy policy-name;
            interface-id <string-value>;
            ipsec-policy ipsec-policy;
        }
    }
}
```

NOTE: For dynamic peers, the Junos OS supports the IKE main mode with either the preshared key method of authentication or an IKE access profile that uses a local digital certificate.

- In preshared key mode, the IP address is used to identify a tunnel peer to get the preshared key information. The `client` value `*` (wildcard) means that configuration within this profile is valid for all dynamic peers terminating within the service set accessing this profile.
- In digital certificate mode, the IKE policy defines which remote identification values are allowed.

The following statements make up the IKE profile:

- `allowed-proxy-pair`—During phase 2 IKE negotiation, the remote peer supplies its network address (`remote`) and its peer's network address (`local`). Since multiple dynamic tunnels are authenticated through the same mechanism, this statement must include the list of possible combinations. If the dynamic peer does not present a valid combination, the phase 2 IKE negotiation fails.

  By default, `remote 0.0.0.0/0 local 0.0.0.0/0` is used if no values are configured. Both IPv4 and IPv6 address formats are supported in this configuration, but there are no default IPv6 addresses. You must specify even `0::0/0`.

- `pre-shared-key`—Key used to authenticate the dynamic peer during IKE phase 1 negotiation. This key is known to both ends through an out-of-band secure mechanism. You can configure the value either in `hexadecimal` or `ascii-text` format. It is a mandatory value.

- `ike-policy`—Policy that defines the remote identification values corresponding to the allowed dynamic peers; can contain a wildcard value `any-remote-id` for use in dynamic endpoint configurations only.

- `interface-id`—Interface identifier, a mandatory attribute used to derive the logical services interface information for the session.

- `ipsec-policy`—Name of the IPsec policy that defines the IPsec policy information for the session. You define the IPsec policy at the `[edit services ipsec-vpn ipsec policy` *`policy-name`*`]` hierarchy level. If no policy is set, any policy proposed by the dynamic peer is accepted.

Referencing the IKE Access Profile in a Service Set

To complete the configuration, you need to reference the IKE access profile configured at the `[edit access]` hierarchy level. To do this, include the `ike-access-profile` statement at the `[edit services service-set` *`name`* `ipsec-vpn-options]` hierarchy level:

```
[edit services service-set name]
ipsec-vpn-options {
    local-gateway address;
    ike-access-profile profile-name;
}
next-hop-service {
    inside-service-interface interface-name;
    outside-service-interface interface-name;
}
```

The `ike-access-profile` statement must reference the same name as the `profile` statement you configured for IKE access at the `[edit access]` hierarchy level. You can reference only one access profile in each service set. This profile is used to negotiate IKE and IPsec security associations with dynamic peers only.

All interfaces referenced by the inside-service-interface statement within a service set must belong to the same VRF instance.

Configuring the Interface Identifier

You can configure an interface identifier for a group of dynamic peers, which specifies which adaptive services logical interface(s) take part in the dynamic IPsec negotiation. By assigning the same interface identifier to multiple logical interfaces, you can create a pool of interfaces for this purpose. To configure an interface identifier, include the ipsec-interface-id statement and the dedicated or shared statement at the [edit interfaces *interface-name* unit *logical-unit-number* dial-options] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number dial-options]
ipsec-interface-id identifier;
(dedicated | shared);
```

Specifying the interface identifier in the dial-options statement makes this logical interface part of the pool identified by the ipsec-interface-id statement.

NOTE: Only one interface identifier can be specified at a time. You can include the ipsec-interface-id statement or the l2tp-interface-id statement, but not both.

If you configure shared mode, it enables one logical interface to be shared across multiple tunnels. The dedicated statement specifies that the logical interface is used in a dedicated mode, which is necessary when you are configuring an IPsec link-type tunnel. You must include the dedicated statement when you specify an ipsec-interface-id value.

Default IKE and IPsec Proposals

The software includes implicit default IKE and IPsec proposals to match the proposals sent by the dynamic peers. The values are shown in Table 25 on page 823 ; if more than one value is shown, the first value is the default.

NOTE: RSA certificates are not supported with dynamic endpoint configuration.

Table 25: Default IKE and IPsec Proposals for Dynamic Negotiations

Statement Name	Values
Implicit IKE Proposal	
authentication-method	pre-shared keys
dh-group	group1, group2, group5, group14
authentication-algorithm	sha1, md5, sha-256
encryption-algorithm	3des-cbc, des-cbc, aes-128, aes-192, aes-256
lifetime-seconds	3600 seconds
Implicit IPsec Proposal	
protocol	esp, ah, bundle
authentication-algorithm	hmac-sha1-96, hmac-md5-96
encryption-algorithm	3des-cbc, des-cbc, aes-128, aes-192, aes-256
lifetime-seconds	28,800 seconds (8 hours)

Distributing Endpoint IPsec Tunnels Among Services Interfaces

Starting in Junos OS Release 16.2R1, you can distribute IPsec tunnels with dynamic endpoints among multiple MS-MICs or among multiple service PICs of an MS-MPC. You configure tunnel distribution by configuring a next-hop IPsec service set for each service PIC’s multiservices (ms-) interface. Starting in Junos OS Release 17.1R1, you can also distribute IPsec tunnels with dynamic endpoints among aggregated multiservices (AMS) interfaces of MS-MICs or MS-MPCs by configuring a next-hop IPsec service set for each AMS interface.

You can later add service PIC hardware to the MX Series router and include the service PIC in the tunnel distribution by simply adding another service set, without needing to change the configuration of the IPsec peers.

To configure tunnel distribution, perform the following steps when configuring dynamic endpoint IPsec tunnels:

- Configure a next-hop IPsec service set for each services interface or AMS interface used by the dynamic endpoint IPsec tunnel (see "Referencing the IKE Access Profile in a Service Set" on page 821). All of the service sets must:
  - Use the same type of services interface—either multiservices (ms-) interfaces or AMS (ams-) interfaces.
  - Have an interface in the `outside-service` statement that is in the same VPN routing and forwarding (VRF) instance as the interfaces in the other service sets.
  - Have the same `local-gateway` IP address.
  - Have the same `ike-access-profile` name.
- When configuring the interface identifier (see "Configuring the Interface Identifier" on page 822), the `ipsec-interface-id` *`identifier`* must be configured:
  - Only under interfaces that appear in the `inside-service-set` statements of the service sets.
  - With `dedicated` for all the interfaces, or with `shared` for all the interfaces.
  - Under no more than one shared unit of an interface.
  - Only under interfaces configured with `service-domain inside`.
  - Only under interfaces that are in the same VRF.

RELATED DOCUMENTATION

Configuring IKE Policies | 681

Configuring IPsec Rules | 723

Configuring IKE Proposals | 675

Configuring IPsec Proposals | 688

Configuring Security Associations | 645

Example: Configuring Dynamically Assigned Policy Based Tunnels

IN THIS SECTION

- Requirements | 825
- Overview and Topology | 825
- Configuration | 826
- Verification | 832

This example shows how to configure dynamically assigned policy-based tunnels and contains the following sections.

Requirements

This example uses the following hardware and software components:

- Three M Series, MX Series or T Series routers.
- Junos OS Release 9.4 or later.

Overview and Topology

IN THIS SECTION

- Topology | 826

An IPsec policy for dynamic endpoints defines a combination of security parameters (IPsec proposals) used during IPsec negotiation between dynamic peer security gateways, in which the remote ends of tunnels do not have a statically assigned IP address.

A policy based VPN is a configuration with a specific VPN tunnel referenced in a policy which acts as a Tunnel. You use a Policy-based VPN if the remote VPN device is a non-Juniper device and if you must access only one subnet or one network at the remote site, across the VPN.

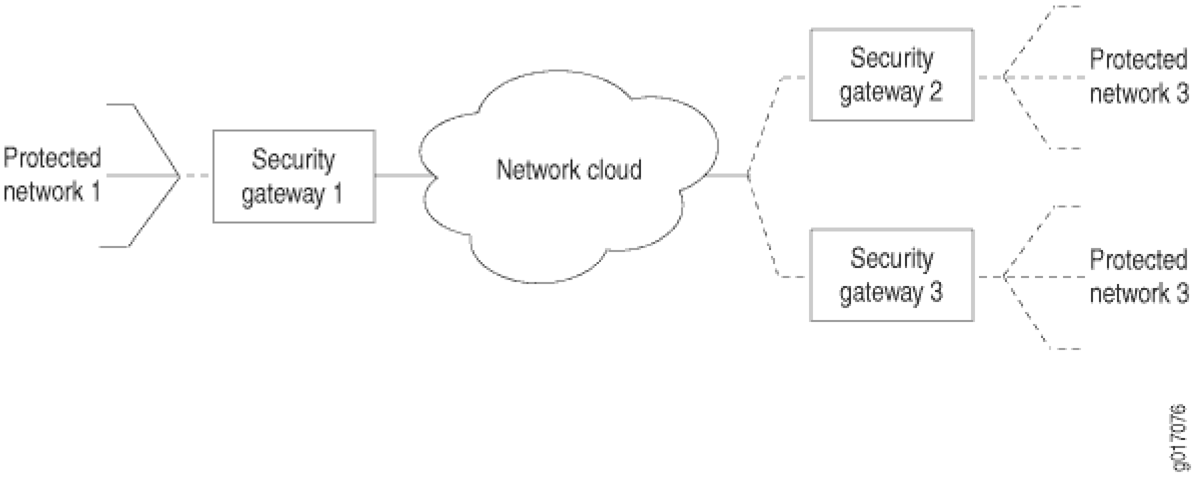
This example explains the IPsec dynamic endpoint tunneling topology as shown in Figure 42 on page 826 .

Before you configure dynamically assigned tunnels, be sure you have:

- A local network N-1 connected to a security gateway SG-1. The exit points must have a Juniper Networks router to terminate the static and dynamic peer endpoints. The tunnel termination address on SG-1 is 10.1.1.1 and the local network address is 172.16.1.0/24.
- Two remote peer routers that obtain addresses from an ISP pool and run an RFC-compliant IKE. The remote network N-2 has the address 172.16.2.0/24 and is connected to the security gateway SG-2 with the tunnel termination address 10.2.2.2. The remote network N-3 has the address 172.16.3.0/24 and is connected to the security gateway SG-3 with the tunnel termination address 10.3.3.3.

Topology

Figure 42: IPsec Dynamic Endpoint Tunneling Topology



Configuration

IN THIS SECTION

- CLI Quick Configuration | 827
- Configuring a Next-Hop SG1 Service-Set | 828
- Results | 830

To configure dynamically assigned policy based tunnels, perform these tasks:

NOTE: The interface types shown in this example are for indicative purpose only. For example, you can use `so-` interfaces instead of `ge-` and `sp-` instead of `ms-`.

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI, at the [edit] hierarchy level, of SG1 router.

Configuring Interfaces

```
set interfaces ms-0/0/0 unit 0 family inet
set interfaces ms-0/0/0 unit 1 family inet
set interfaces ms-0/0/0 unit 1 service-domain inside
set interfaces ms-0/0/0 unit 1 dial-options ipsec-interface-id demo-ipsec-interface-id
set interfaces ms-0/0/0 unit 1 dial-options  shared
set interfaces ms-0/0/0 unit 2 family inet
set interfaces ms-0/0/0 unit 2 service-domain outside
```

Configuring Access Profile

```
set access profile demo-access-profile client *  ike allowed-proxy-pair remote 172.16.2.0/24
local 172.16.1.0/24
set access profile demo-access-profile client *  ike allowed-proxy-pair remote 172.16.3.0/24
local 172.16.1.0/24
set access profile demo-access-profile client *  ike ascii-text keyfordynamicpeers
set access profile demo-access-profile client *  ike interface-id demo-ipsec-interface-id
```

Configuring Service Set

```
set services service-set demo-service-set next-hop-service inside-service-interface ms-0/0/0.1
set services service-set demo-service-set next-hop-service outside-service-interface ms-0/0/0.2
```

Configuring IPsec Properties

```
set services ipsec-vpn ipsec proposal ipsec_proposal_demo1 protocol esp
set services ipsec-vpn ipsec proposal ipsec_proposal_demo1 authentication-algorithm hmac-sha1-96
set services ipsec-vpn ipsec proposal ipsec_proposal_demo1 encryption-algorithm 3des-cbc
set services ipsec-vpn ipsec policy demo2 perfect-forward-secrecy keys group2
set services ipsec-vpn ipsec policy demo2 proposals ipsec_proposal_demo1
set services ipsec-vpn ike proposal ike_proposal_demo1 authentication-method pre-shared-keys
set services ipsec-vpn ike proposal ike_proposal_demo1 dh-group group2
set services ipsec-vpn ike policy ike_policy_demo1 version 2
set services ipsec-vpn ike policy ike_policy_demo1 proposals ike_proposal_demo1
set services ipsec-vpn ike policy ike_policy_demo1 pre-shared-key ascii-text keyfordemo1
```

Configuring Routing Instances

```
set routing-instances demo-vrf instance-type vrf
set routing-instances demo-vrf ms-0/0/0.1
set routing-instances demo-vrf ms-0/0/0.2
```

Configuring a Next-Hop SG1 Service-Set

Step-by-Step Procedure

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy.

1. Configure the interfaces.

```
[edit interfaces]
user@router1# set interfaces ms-0/0/0 unit 0 family inet
user@router1# set interfaces ms-0/0/0 unit 1 family inet
user@router1# set interfaces ms-0/0/0 unit 1 service-domain inside
user@router1# set interfaces ms-0/0/0 unit 1 dial-options ipsec-interface-id demo-ipsec-
interface-id
user@router1# set interfaces ms-0/0/0 unit 1 dial-options mode shared
user@router1# set interfaces ms-0/0/0 unit 2 family inet
user@router1# set interfaces ms-0/0/0 unit 2 service-domain outside
```

2. Configure the access profile.

```
[edit access]
user@router1# set profile demo-access-profile client *  ike allowed-proxy-pair remote
```

```
172.16.2.0/24 local 172.16.1.0/24
user@router1# set profile demo-access-profile client *  ike ascii-text keyfordynamicpeers
user@router1# set profile demo-access-profile client *  ike interface-id demo-ipsec-interface-
id
```

3. Configure the services set.

```
[edit services]
user@router1# set service-set demo-service-set next-hop-service inside-service-interface
ms-0/0/0.1
user@router1# set service-set demo-service-set next-hop-service outside-service-interface
ms-0/0/0.2
```

4. Configure the IPsec properties.

```
[edit services ipsec-vpn]
user@router1#set ipsec proposal ipsec_proposal_demo1 protocol esp
user@router1#set ipsec proposal ipsec_proposal_demo1 authentication-algorithm hmac-sha1-96
user@router1#set ipsec proposal ipsec_proposal_demo1 encryption-algorithm 3des-cbc
user@router1#set ipsec policy demo2 perfect-forward-secrecy keys group2
user@router1#set ipsec policy demo2 proposals ipsec_proposal_demo1
user@router1#set ike proposal ike_proposal_demo1 authentication-method pre-shared-keys
user@router1#set ike proposal ike_proposal_demo1 dh-group group2
user@router1#set ike policy ike_policy_demo1 version 2
user@router1#set ike policy ike_policy_demo1 proposals ike_proposal_demo1
user@router1#set ike policy ike_policy_demo1 pre-shared-key ascii-text keyfordemo1
```

5. Configure the routing instances.

```
[edit routing-instances]
user@router1# set demo-vrf instance-type vrf
user@router1# set demo-vrf ms-0/0/0.1
user@router1# set demo-vrf ms-0/0/0.2
```

Results

From configuration mode of Router 1, confirm your configuration by entering the `show interfaces`, `show access`, and `show services` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
interfaces {
    ms-0/0/0 {
        unit 0 {
            family inet;
        }
        unit 1 {
            family inet;
            service-domain inside;
            dial-options {
                ipsec-interface-id demo-ipsec-interface-id;
                mode shared;
            }
        }
        unit 2 {
            family inet;
            service-domain outside;
        }
    }
}
access {
    profile demo-access-profile client * {
        ike {
            allowed-proxy-pair {
                remote 172.16.2.0/24 local 172.16.1.0/24; #Set for Network 2 connected to 
Network 1
                remote 172.16.3.0/24 local 172.16.1.0/24; #Set for Network 3 connected to 
Network 1
            }
            pre-shared-key {
                ascii-text keyfordynamicpeers;
            }
            interface-id demo-ipsec-interface-id;
        }
    }
}
services {
    service-set demo-service-set {
        next-hop-service {
            inside-service-interface ms-0/0/0.1;
            outside-service-interface ms-0/0/0.2;
        }
        ipsec-vpn-options {
            local-gateway 10.1.1.1;
            ike-access-profile demo-access-profile;
        }
    }
    ipsec-vpn {
        ipsec {
            proposal ipsec_proposal_demo1 {
                protocol esp;
                authentication-algorithm hmac-sha1-96;
                encryption-algorithm 3des-cbc;
            }
            policy demo2 {
                perfect-forward-secrecy {
                    keys group2;
                }
                proposals ipsec_proposal_demo1;
            }
        }
        ike {
            proposal ike_proposal_demo1 {
                authentication-method pre-shared-keys;
                dh-group group2;
            }
            policy ike_policy_demo1 {
                version 2;
                proposals ike_proposal_demo1;
                pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
            }
        }
    }
}
routing-instances {
    demo-vrf {
        instance-type vrf;
        interface ms-0/0/0.1;
        interface ms-0/0/0.2;
    }
}
```

Verification

IN THIS SECTION

- Verifying That the Next-Hop SG1 Service Set with Policy-Based Tunnels Is Created | **832**

Verifying That the Next-Hop SG1 Service Set with Policy-Based Tunnels Is Created

Purpose

Verify that the next-hop SG1 service set with policy-based tunnels is created.

Action

From operational mode, enter the `show route` command.

```
user@router1> show route
demo-vrf.inet.0: .... # Routing instance
172.11.0.0/24 *[Static/1]..
    > via ms-0/0/0.1
    172.12.0.0/24 *[Static/1]..
> via ms-0/0/0.1
```

From operational mode, enter the `show services ipsec-vpn ipsec security-associations detail`

```
user@router1>show services ipsec-vpn ipsec security-associations detail
rule: junos-dynamic-rule-0
term: term-0
local-gateway-address : 10.1.1.1 #Tunnel termination address on SG-1
remote-gateway-address: 10.2.2.2 #Tunnel termination address on SG-2
source-address : 0.0.0.0/0
destination-address : 0.0.0.0/0
ipsec-inside-interface: ms-0/0/0.1
term: term-1
```

```
local-gateway-address : 10.1.1.1 #Tunnel termination address on SG-1
remote-gateway-address: 10.3.3.3 #Tunnel termination address on SG-3
source-address : 0.0.0.0/0
destination-address : 0.0.0.0/0
IPsec Properties
ipsec-inside-interface: ms-0/0/0.1
match-direction: input
```

Meaning

The `show services ipsec-vpn ipsec security-associations detail` command output shows the properties that you configured.

SEE ALSO

- Understanding Junos VPN Site Secure | 623
- Configuring Security Associations | 645
- Configuring IPsec Policies | 694
- Configuring IKE Policies | 681

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
17.1	Starting in Junos OS Release 17.1R1, you can also distribute IPsec tunnels with dynamic endpoints among aggregated multiservices (AMS) interfaces of MS-MICs or MS-MPCs by configuring a next-hop IPsec service set for each AMS interface.
16.2	Starting in Junos OS Release 16.2R1, you can distribute IPsec tunnels with dynamic endpoints among multiple MS-MICs or among multiple service PICs of an MS-MPC. You configure tunnel distribution by configuring a next-hop IPsec service set for each service PIC's multiservices (ms-) interface.

8

PART

CoS on Services Cards

CoS on Services Cards | 835

Class of Service on Link Services Interfaces | 847

CHAPTER 32

CoS on Services Cards

IN THIS CHAPTER

- Class of Service on Services Interfaces | **835**

Class of Service on Services Interfaces

IN THIS SECTION

- Class of Service Overview | **835**
- Restrictions and Cautions for CoS Configuration on Services Interfaces | **836**
- Configuring CoS Rules | **837**
- Configuring CoS Rule Sets | **843**
- Examples: Configuring CoS on Services Interfaces | **844**

Class of Service Overview

The CoS configuration available for the M Series and MX Series-based service cards enables you to configure Differentiated Services (DiffServ) code point (DSCP) marking and forwarding-class assignment for packets transiting the service cards-service PICs. The M Series and MX Series-based service cards include Multiservices PIC, MS-MIC, MS-MPC, MS-DPC, and Adaptive Services PIC. You can configure the CoS service alongside the stateful firewall and NAT services, using a similar rule structure. The component structures are described in detail in the Class of Service User Guide (Routers and EX9200 Switches).

Standards for Differentiated Services are described in the following documents:

- RFC 2474, *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*
- RFC 2475, *An Architecture for Differentiated Services*

NOTE: CoS BA classification is not supported on services interfaces. The CoS configuration is available only for NAT and stateful firewall services. The CoS configuration does not work with other services that run on the service cards such as IPsec.

Restrictions and Cautions for CoS Configuration on Services Interfaces

The following restrictions and cautions apply to CoS configuration on services interfaces:

- You must configure at least one stateful firewall rule or NAT rule on the service set. Otherwise, CoS does not work.

- The services interfaces do not support scheduling, only DiffServ marking and queue assignment. You must configure scheduling at the `[edit class-of-service]` hierarchy level on the output interface or fabric.

- In the default configuration, queues 1 and 2 receive 0 percent bandwidth. If packets will be assigned to these queues, you must configure a scheduling map.

- You must issue a `commit full` command before using custom forwarding-class names in the configuration.

- Only the Junos standard DiffServ names can be used in the configuration. Custom names are not recognized.

- On M Series routers, you can configure *rewrite rules* that change packet headers and attach the rules to output interfaces. These rules might overwrite the DSCP marking configured on a MultiServices PIC. It is important to keep this adverse effect in mind and use care when creating system-wide configurations.

  For example, knowing that the MultiServices PIC can mark packets with any ToS or DSCP value and the output interface is restricted to only eight DSCP values, rewrite rules on the output interface condense the mapping from 64 to 8 values with overall loss of granularity. In this case, you have the following options:

  - Remove the rewrite rules from the output interface.

  - Configure the output interface to include the most important mappings.

Configuring CoS Rules

IN THIS SECTION

- Configuring Match Direction for CoS Rules | 838
- Configuring Match Conditions In CoS Rules | 839
- Configuring Actions in CoS Rules | 840
- Configuring CoS Session Creation When Packet Received in Non-Matching Direction | 842
- Example: Configuring CoS Rules | 843

To configure a CoS rule, include the `rule` *`rule-name`* statement at the `[edit services cos]` hierarchy level:

```
[edit services cos]
rule rule-name {
    match-direction (input | output | input-output);
    term term-name {
        from {
            application-sets set-name;
            applications [ application-names ];
            destination-address address;
            destination-prefix-list list-name <except>;
            source-address address;
            source-prefix-list list-name <except>;
        }
        then {
            application-profile profile-name;
            dscp (alias | bits);
            forwarding-class class-name;
            syslog;
            reflexive; | revert; | reverse {
                application-profile profile-name;
                dscp (alias | bits);
                forwarding-class class-name;
                syslog;
            }
        }
    }
}
```

Each CoS rule consists of a set of terms, similar to a filter configured at the `[edit firewall]` hierarchy level. A term consists of the following:

- `from` statement—Specifies the match conditions and applications that are included and excluded.
- `then` statement—Specifies the actions and action modifiers to be performed by the router software.

Apply the CoS rule to a service set at the `[edit services]` hierarchy level:

```
[edit services]
service-set service-set-name {
    cos-rules [cos-rule-name];
    }
```

The following sections explain how to configure the components of CoS rules:

Configuring Match Direction for CoS Rules

Each rule must include a `match-direction` statement that specifies the direction in which the rule match is applied. To configure where the match is applied, include the `match-direction` statement at the `[edit services cos rule rule-name]` hierarchy level:

```
match-direction (input | output | input-output);
```

If you configure `match-direction input-output`, bidirectional rule creation is allowed.

The match direction is used with respect to the traffic flow through the Multiservices PIC, MS-MIC, or MS-MPC. When a packet is sent to the PIC, direction information is carried along with it.

With an interface service set, packet direction is determined by whether a packet is entering or leaving the interface on which the service set is applied.

With a next-hop service set, packet direction is determined by the interface used to route the packet to the Multiservices PIC, MS-MIC, or MS-MPC. If the inside interface is used to route the packet, the packet direction is input. If the outside interface is used to direct the packet to the Multiservices PIC, MS-MIC, or MS-MPC, the packet direction is output. For more information on inside and outside interfaces, see "Configuring Service Sets to be Applied to Services Interfaces" on page 10 .

On the Multiservices PIC, MS-MIC, or MS-MPC, a flow lookup is performed. If no flow is found, rule processing is performed. All rules in the service set are considered. During rule processing, the packet direction is compared against rule directions. Only rules with direction information that matches the packet direction are considered.

Configuring Match Conditions In CoS Rules

To configure CoS match conditions, include the `from` statement at the `[edit services cos rule` *`rule-name`* `term` *`term-name`*`]` hierarchy level:

```
from {
    application-sets set-name;
    applications [ application-names ];
    destination-address address;
    destination-prefix-list list-name <except>;
    source-address address;
    source-prefix-list list-name <except>;
}
```

The source address and destination address can be either IPv4 or IPv6. You can use either the source address or the destination address as a match condition, in the same way that you would configure a firewall filter; for more information, see the Routing Policies, Firewall Filters, and Traffic Policers User Guide.

Alternatively, you can specify a list of source or destination prefixes by configuring the `prefix-list` statement at the `[edit policy-options]` hierarchy level and then including either the `destination-prefix-list` or `source-prefix-list` statement in the CoS rule. For an example, see "Examples: Configuring Stateful Firewall Rules" on page 557 .

If you omit the `from` term, the router accepts all traffic and the default protocol handlers take effect:

- User Datagram Protocol (UDP), Transmission Control Protocol (TCP), and Internet Control Message Protocol (ICMP) create a bidirectional flow with a predicted reverse flow.

- IP creates a unidirectional flow.

You can also include application protocol definitions you have configured at the `[edit applications]` hierarchy level; for more information, see "Configuring Application Properties" on page 508 .

- To apply one or more specific application protocol definitions, include the `applications` statement at the `[edit services cos rule` *`rule-name`* `term` *`term-name`* `from]` hierarchy level.

- To apply one or more sets of application protocol definitions you have defined, include the `application-sets` statement at the `[edit services cos rule` *`rule-name`* `term` *`term-name`* `from]` hierarchy level.

NOTE: If you include one of the statements that specifies application protocols, the router derives port and protocol information from the corresponding configuration at the [edit applications] hierarchy level; you cannot specify these properties as match conditions.

Configuring Actions in CoS Rules

To configure CoS actions, include the then statement at the [edit services cos rule *rule-name* term *term-name*] hierarchy level:

```
[edit services cos rule rule-name term term-name]
then {
    application-profile profile-name;
    dscp (alias | bits);
    forwarding-class class-name;
    syslog;
    reflexive; | revert; | reverse {
        application-profile profile-name;
        dscp (alias | bits);
        forwarding-class class-name;
        syslog;
    }
}
```

The principal CoS actions are as follows:

- dscp—Causes the packet to be marked with the specified DiffServ code point (DSCP) value or alias.
- forwarding-class—Causes the packet to be assigned to the specified forwarding class.

For detailed information about DSCP values and forwarding classes, see "Examples: Configuring CoS on Services Interfaces" on page 844 or the Class of Service User Guide (Routers and EX9200 Switches).

You can optionally set the configuration to record information in the system logging facility by including the syslog statement at the [edit services cos rule *rule-name* term *term-name* then] hierarchy level. This statement overrides any syslog setting included in the service set or interface default configuration.

For information about some additional CoS actions, see the following sections:

Configuring Application Profiles for Use as CoS Rule Actions

You can optionally define one or more application profiles for inclusion in CoS actions. To configure application profiles, include the `application-profile` statement at the `[edit services cos]` hierarchy level:

```
[edit services cos]
application-profile profile-name {
    ftp {
        data {
            dscp (alias | bits);
            forwarding-class class-name;
        }
    }
    sip {
        video {
            dscp (alias | bits);
            forwarding-class class-name;
        }
        voice {
            dscp (alias | bits);
            forwarding-class class-name;
        }
    }
}
```

The `application-profile` statement includes two main components and three traffic types: `ftp` with the `data` traffic type and `sip` with the `video` and `voice` traffic types. You can set the appropriate `dscp` and `forwarding-class` values for each component within the application profile.

NOTE: The `ftp` and `sip` statements are not supported on Juniper Network MX Series 3D Universal Edge Routers.

You can apply the application profile to a CoS configuration by including it at the `[edit services cos rule` *`rule-name`* `term` *`term-name`* `then]` hierarchy level.

Configuring Reflexive, Revert, and Reverse CoS Rule Actions

CoS services are unidirectional. It might be necessary to specify different treatments for flows in opposite directions.

Regardless of whether a packet matches the input, output or input-output direction, flows in both directions are created. A forward, reverse, or forward-and-reverse CoS action is associated with each flow. Bear in mind that the flow in the reverse direction might end up having a CoS action associated with it that you have not specifically configured.

To control the direction in which service is applied, as distinct from the direction in which the rule match is applied, you can configure the (`reflexive | revert | reverse`) statement at the `[edit services cos rule rule-name term term-name then]` hierarchy level:

```
[edit services cos rule rule-name term term-name then]
reflexive; | revert; | reverse {
    application-profile profile-name;
    dscp (alias | bits);
    forwarding-class class-name;
    syslog;
}
```

The three actions are mutually exclusive:

- `reflexive` causes the CoS rule actions to be applied to flows in the reverse direction as well as to flows in the matching direction.
- Starting with Junos OS Release 16.1R5 and Junos OS Release 17.4R1, `revert` stores the DSCP and forwarding class of a packet that is received in the match direction of the rule and then applies that DSCP and forwarding class to packets that are received in the reverse direction of the same session.
- `reverse` allows you to define the CoS behavior for flows in the reverse direction.

If you omit the statement, data flows inherit the CoS behavior of the forward control flow.

Configuring CoS Session Creation When Packet Received in Non-Matching Direction

Starting with Junos OS Release 16.1R5 and Junos OS Release 17.4R1, you can configure a service set to create a CoS session even if a packet is first received in the wrong match direction for a CoS rule that is assigned to the service set. This results in the CoS rule values being applied as soon as a packet in the correct match direction is received. To configure this capability, include the `match-rules-on-reverse-flow` at the `[edit services service-set service-set-name cos-options]` hierarchy level:

```
[edit services service-set service-set-name cos-options]
match-rules-on-reverse-flow;
```

Example: Configuring CoS Rules

The following example shows a CoS configuration containing two rules, one for input matching on a specified application set and the other for output matching on a specified source address:

```
[edit services]
cos {
    rule my-cos-rule {
        match-direction input-output;
        term term1 {
            from {
                source-address 10.1.3.2/32;
                application-set sip;
            }
            then {
                dscp ef;
                syslog;
            }
        }
        term term2 {
            from {
                destination-address 10.2.3.2;
                applications http;
            }
            then {
                dscp af21;
            }
        }
    }
}
```

Configuring CoS Rule Sets

The `rule-set` statement defines a collection of CoS rules that determine what actions the router software performs on packets in the data stream. You define each rule by specifying a rule name and configuring terms. Then you specify the order of the rules by including the `rule-set` statement at the `[edit services cos]` hierarchy level with a `rule` statement for each rule:

```
rule-set rule-set-name {
    rule rule-name;
}
```

The router software processes the rules in the order in which you specify them in the configuration. If a term in a rule matches the packet, the router performs the corresponding action and the rule processing stops. If no term in a rule matches the packet, processing continues to the next rule in the rule set. If none of the rules matches the packet, the packet is dropped by default.

Examples: Configuring CoS on Services Interfaces

To make settings consistent across Juniper Networks routers, you configure many CoS settings at the `[edit class-of-service]` hierarchy level to be used on services interfaces. When you commit this configuration along with what you configure at the `[edit services cos]` hierarchy level, these properties are applied to the Multiservices PIC, MS-MIC, or MS-MPC.

The following configuration examples at the `[edit class-of-service]` hierarchy level can be applied on services interfaces. For more information, see the Class of Service User Guide (Routers and EX9200 Switches).

NOTE: The first two configurations, mapping forwarding-class name to forwarding-class ID and mapping forwarding-class name to queue number, are mutually exclusive.

Mapping Forwarding-Class Name to Forwarding-Class ID

Map forwarding-class names to forwarding-class IDs:

```
[edit class-of-service]
forwarding-classes {
    forwarding-class fc0 0;
    forwarding-class fc1 0;
    forwarding-class fc2 1;
    forwarding-class fc3 1;
    forwarding-class fc4 2;
    forwarding-class fc5 2;
    forwarding-class fc6 3;
    forwarding-class fc7 3;
    forwarding-class fc8 4;
    forwarding-class fc9 4;
    forwarding-class fc10 5;
    forwarding-class fc11 5;
    forwarding-class fc12 6;
    forwarding-class fc13 6;
    forwarding-class fc14 7;
    forwarding-class fc15 7;
}
```

Mapping Forwarding-Class Name to Queue Number

Map forwarding-class names to queue numbers:

```
[edit class-of-service]
forwarding-classes {
    queue 0 be;
    queue 1 ef;
    queue 2 af;
    queue 3 nc;
    queue 4 ef1;
    queue 5 ef2;
    queue 6 af1;
    queue 7 nc1;
}
```

Mapping Diffserv Code Point Aliases to DSCP Bits

Map alias names to DSCP bit values. The aliases then can be used instead of the DSCP bits in adaptive services configurations.

```
[edit class-of-service]
code-point-aliases {
    (dscp | dscp-ipv6 | exp | ieee-802.1 | inet-precedence) {
        alias | bits;
    }
}
```

Here is an example:

```
code-point-aliases {
    dscp {
        my1 110001;
        my2 101110;
        be 000001;
        cs7 110000;
    }
}
```

SEE ALSO

Class of Service Overview | 835

Restrictions and Cautions for CoS Configuration on Services Interfaces | 836

Configuring CoS Rules | 837

Configuring CoS Rule Sets | 843

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
16.1R5	Starting with Junos OS Release 16.1R5 and Junos OS Release 17.4R1, `revert` stores the DSCP and forwarding class of a packet that is received in the match direction of the rule and then applies that DSCP and forwarding class to packets that are received in the reverse direction of the same session.
16.1R5	Starting with Junos OS Release 16.1R5 and Junos OS Release 17.4R1, you can configure a service set to create a CoS session even if a packet is first received in the wrong match direction for a CoS rule that is assigned to the service set.

CHAPTER 33

Class of Service on Link Services Interfaces

IN THIS CHAPTER

- Class of Service on Link Services Interfaces | 847

Class of Service on Link Services Interfaces

IN THIS SECTION

- Link Services Configuration for Junos Interfaces | 847
- Configuring CoS Scheduling Queues on Logical LSQ Interfaces | 849
- Configuring CoS Fragmentation by Forwarding Class on LSQ Interfaces | 853
- Configuring Link Services and CoS on Services PICs | 856
- Oversubscribing Interface Bandwidth on LSQ Interfaces | 860
- Configuring Guaranteed Minimum Rate on LSQ Interfaces | 866

Link Services Configuration for Junos Interfaces

This topic provides links to topics explaining link services configuration for the following interface types:

- For information about configuring LSQ interface redundancy across multiple routers using SONET APS interfaces, see "Configuring LSQ Interface Redundancy Across Multiple Routers Using SONET APS" on page 921

- For information about configuring LSQ interface redundancy in a single router using SONET APS interfaces, see "Configuring LSQ Interface Redundancy in a Single Router Using SONET APS" on page 921

- For information about configuring LSQ interface redundancy in a single router using Virtual Interfaces, see "Configuring LSQ Interface Redundancy in a Single Router Using Virtual Interfaces" on page 921
- For information about configuring CoS scheduling queues on Logical LSQ interfaces, see "Configuring CoS Scheduling Queues on Logical LSQ Interfaces" on page 849
- For information about configuring CoS fragmentation by forwarding class on LSQ interfaces, see"Configuring CoS Fragmentation by Forwarding Class on LSQ Interfaces" on page 853
- For information about reserving bundle bandwidth for Link-Layer overhead on LSQ interfaces, see "Reserving Bundle Bandwidth for Link-Layer Overhead on LSQ Interfaces" on page 939
- For information about oversubscribing interface bandwidth on LSQ interfaces, see "Oversubscribing Interface Bandwidth on LSQ Interfaces" on page 860
- For information about configuring guaranteed minimum rate on LSQ interfaces, see "Configuring Guaranteed Minimum Rate on LSQ Interfaces" on page 866
- For information about configuring link services and CoS on services PICs, see"Configuring Link Services and CoS on Services PICs" on page 856
- For information about configuring LSQ interfaces as NxT1 or NxE1 bundles using MLPPP, see"Configuring LSQ Interfaces as NxT1 or NxE1 Bundles Using MLPPP" on page 939
- For information about configuring LSQ interfaces as NxT1 or NxE1 bundles using FRF.16, see"Configuring LSQ Interfaces as NxT1 or NxE1 Bundles Using FRF.16" on page 939
- For information about configuring LSQ interfaces for single fractional T1 or E1 interfaces using MLPPP and LFI, see "Configuring LSQ Interfaces for Single Fractional T1 or E1 Interfaces Using MLPPP and LFI" on page 939
- For information about configuring LSQ interfaces for single fractional T1 or E1 interfaces using FRF.12, see "Configuring LSQ Interfaces for Single Fractional T1 or E1 Interfaces Using FRF.12" on page 939
- For information about configuring LSQ interfaces as NxT1 or NxE1 bundles using FRF.15, see "Configuring LSQ Interfaces as NxT1 or NxE1 Bundles Using FRF.15" on page 939
- For information about configuring LSQ interfaces for T3 links configured for compressed RTP over MLPPP, see "Configuring LSQ Interfaces for T3 Links Configured for Compressed RTP over MLPPP" on page 939
- For information about configuring LSQ interfaces as T3 or OC3 bundles using FRF.12, see "Configuring LSQ Interfaces as T3 or OC3 Bundles Using FRF.12" on page 939

- For information about configuring LSQ interfaces for ATM2 IQ interfaces using MLPPP, see "Configuring LSQ Interfaces for ATM2 IQ Interfaces Using MLPPP" on page 939

SEE ALSO

Layer 2 Service Package Capabilities and Interfaces | 921

Configuring CoS Scheduling Queues on Logical LSQ Interfaces

IN THIS SECTION

- Configuring Scheduler Buffer Size | 851
- Configuring Scheduler Priority | 851
- Configuring Scheduler Shaping Rate | 851
- Configuring Drop Profiles | 852

For link services IQ (`lsq-`) interfaces, you can specify a scheduler map for each logical unit. A logical unit represents either an MLPPP bundle or a DLCI configured on a FRF.16 bundle. The scheduler is applied to the traffic sent to an AS or Multiservices PIC running the Layer 2 link services package.

If you configure a scheduler map on a bundle, you must include the `per-unit-scheduler` statement at the `[edit interfaces lsq-`*`fpc/pic/port`*`]` hierarchy level. If you configure a scheduler map on an FRF.16 DLCI, you must include the `per-unit-scheduler` statement at the `[edit interfaces lsq-`*`fpc/pic/port`*`:`*`channel`*`]` hierarchy level. For more information, see the Class of Service User Guide (Routers and EX9200 Switches).

If you need latency guarantees for multiclass or LFI traffic, you must use channelized IQ PICs for the constituent links. With non-IQ PICs, because queueing is not done at the channelized interface level on the constituent links, latency-sensitive traffic might not receive the type of service that it should. Constituent links from the following PICs support latency guarantees:

- Channelized E1 IQ PIC
- Channelized OC3 IQ PIC
- Channelized OC12 IQ PIC
- Channelized STM1 IQ PIC
- Channelized T3 IQ PIC

For scheduling queues on a logical interface, you can configure the following scheduler map properties at the `[edit class-of-service schedulers]` hierarchy level:

- `buffer-size`—The queue size; for more information, see "Configuring Scheduler Buffer Size" on page 851 .
- `priority`—The transmit priority (low, high, strict-high); for more information, see "Configuring Scheduler Priority" on page 851 .
- `shaping-rate`—The subscribed transmit rate; for more information, see "Configuring Scheduler Shaping Rate" on page 851 .
- `drop-profile-map`—The random early detection (RED) drop profile; for more information, see "Configuring Drop Profiles" on page 852 .

When you configure MLPPP and FRF.12 on M Series and T Series routers, you should configure a single scheduler with non-zero percent transmission rates and buffer sizes for queues 0 through 3, and assign this scheduler to the link services IQ interface (`lsq`) and to each constituent link.

When you configure FRF.16 on M Series and T Series routers, you can assign a single scheduler map to the link services IQ interface (`lsq`) and to each link services IQ DLCI, or you can assign different scheduler maps to the various DLCIs of the bundle, as shown in "Example: Configuring an LSQ Interface as an NxT1 Bundle Using FRF.16" on page 939 . For the constituent links of an FRF.16 bundle, you do not need to configure a custom scheduler. Because LFI and multiclass are not supported for FRF.16, the traffic from each constituent link is transmitted from queue 0. This means you should allow most of the bandwidth to be used by queue 0. The default scheduler transmission rate and buffer size percentages for queues 0 through 3 are 95, 0, 0, and 5 percent, respectively. This default scheduler sends all user traffic to queue 0 and all network-control traffic to queue 3, and therefore it is well suited to the behavior of FRF.16. You can configure a custom scheduler that explicitly replicates the 95, 0, 0, and 5 percent queuing behaviors, and apply it to the constituent links.

NOTE: On T Series and M320 routers, the default scheduler transmission rate and buffer size percentages for queues 0 through 7 are 95, 0, 0, 5, 0, 0, 0, and 0 percent.

For link services IQ interfaces (`lsq`), these scheduling properties work as they do in other PICs, except as noted in the following sections.

NOTE: On T Series and M320 routers, `lsq` interfaces do not support DiffServ code point (DSCP) and DSCP-IPv6 rewrite markers.

Configuring Scheduler Buffer Size

You can configure the scheduler buffer size in three ways: as a temporal value, as a percentage, and as a remainder. On a single logical interface (MLPPP or a FRF.16 DLCI), each queue can have a different buffer size.

If you specify a temporal value, the queuing algorithm starts dropping packets when it queues more than a computed number of bytes. This number is computed by multiplying logical interface speed by the temporal value. For MLPPP bundles, logical interface speed is equal to the bundle bandwidth, which is the sum of constituent link speeds minus link-layer overhead. For MLFR FRF.16 DLCIs, logical interface speed is equal to bundle bandwidth multiplied by the DLCI shaping rate. In all cases, the maximum temporal value is limited to 200 milliseconds.

Buffer size percentages are implicitly converted into temporal values by multiplying the percentage by 200 milliseconds. For example, buffer size specified as `buffer-size percent 20` is the same as a 40-millisecond temporal delay. The link services IQ implementation guarantees 200 milliseconds of buffer delay for all interfaces with T1 and higher speeds. For slower interfaces, it guarantees one second of buffer delay.

The queueing algorithm evenly distributes leftover bandwidth among all queues that are configured with the `buffer-size remainder` statement. The queuing algorithm guarantees enough space in the transmit buffer for two MTU-sized packets.

Configuring Scheduler Priority

The transmit priority of each queue is determined by the scheduler and the forwarding class. Each queue receives a guaranteed amount of bandwidth specified with the scheduler `transmit-rate` statement.

Configuring Scheduler Shaping Rate

You use the shaping rate to set the percentage of total bundle bandwidth that is dedicated to a DLCI. For link services IQ DLCIs, only percentages are accepted, which allows adjustments in response to dynamic changes in bundle bandwidth—for example, when a link goes up or down. This means that absolute shaping rates are not supported on FRF.16 bundles. Absolute shaping rates are allowed for MLPPP and MLFR bundles only.

For scheduling between DLCIs in a MLFR FRF.16 bundle, you can configure a shaping rate for each DLCI. A shaping rate is expressed as a percentage of the aggregate bundle bandwidth. Shaping rate percentages for all DLCIs within a bundle can add up to 100 percent or less. Leftover bandwidth is distributed equally to DLCIs that do not have the `shaping-rate` statement included at the `[edit class-of-service interfaces lsq-`*`fpc/pic/port:channel`*` unit `*`logical-unit-number`*`]` hierarchy level. If none of the DLCIs in an MLFR FRF.16 bundle specify a DLCI scheduler, the total bandwidth is evenly divided across all DLCIs.

NOTE: For FRF.16 bundles on link services IQ interfaces, only shaping rates based on percentage are supported.

Configuring Drop Profiles

You can configure random early detection (RED) on LSQ interfaces as in other CoS scenarios. To configure RED, include one or more drop profiles and attach them to a scheduler for a particular forwarding class. For more information about RED profiles, see the Class of Service User Guide (Routers and EX9200 Switches).

The LSQ implementation performs tail RED. It supports a maximum of 256 drop profiles per PIC. Drop profiles are configurable on a per-queue, per-loss-priority, and per-TCP-bit basis.

You can attach scheduler maps with configured RED drop profiles to any LSQ logical interface: an MLPPP bundle, an FRF.15 bundle, or an FRF.16 DLCI. Different queues (forwarding classes) on the same logical interface can have different associated drop profiles.

The following example shows how to configure a RED profile on an LSQ interface:

```
[edit]
class-of-service {
    drop-profiles {
        drop-low {
            # Configure suitable drop profile for low loss priority
            ...
        }
        drop-high {
            # Configure suitable drop profile for high loss priority
            ...
        }
    }
    scheduler-maps {
        schedmap {
            # Best-effort queue will use be-scheduler
            # Other queues may use different schedulers
            forwarding-class be scheduler be-scheduler;
            ...
        }
    }
    schedulers {
        be-scheduler {
            # Configure two drop profiles for low and high loss priority
            drop-profile-map loss-priority low protocol any drop-profile drop-low;
            drop-profile-map loss-priority high protocol any drop-profile drop-high;
            # Other scheduler parameters (buffer-size, priority,
            # and transmit-rate) are already supported.
            ...
        }
    }
    interfaces {
        lsq-1/3/0.0 {
            # Attach a scheduler map (that includes RED drop profiles)
            # to a LSQ logical interface.
            scheduler-map schedmap;
        }
    }
}
```

NOTE: The RED profiles should be applied only on the LSQ bundles and not on the egress links that constitute the bundle.

SEE ALSO

Layer 2 Service Package Capabilities and Interfaces | 921

Configuring CoS Fragmentation by Forwarding Class on LSQ Interfaces

For link services IQ (`lsq-`) interfaces, you can specify fragmentation properties for specific forwarding classes. Traffic on each forwarding class can be either multilink encapsulated (fragmented and sequenced) or nonencapsulated (hashed with no fragmentation). By default, traffic in all forwarding classes is multilink encapsulated.

When you do not configure fragmentation properties for the queues on MLPPP interfaces, the fragmentation threshold you set at the `[edit interfaces` *`interface-name`* `unit` *`logical-unit-number`* `fragment-threshold]` hierarchy level is the fragmentation threshold for all forwarding classes within the MLPPP interface. For MLFR FRF.16 interfaces, the fragmentation threshold you set at the `[edit interfaces` *`interface-name`* `mlfr-uni-nni-bundle-options fragment-threshold]` hierarchy level is the fragmentation threshold for all forwarding classes within the MLFR FRF.16 interface.

If you do not set a maximum fragment size anywhere in the configuration, packets are still fragmented if they exceed the smallest maximum transmission unit (MTU) or maximum received reconstructed unit (MRRU) of all the links in the bundle. A nonencapsulated flow uses only one link. If the flow exceeds a single link, then the forwarding class must be multilink encapsulated, unless the packet size exceeds the MTU/MRRU.

Even if you do not set a maximum fragment size anywhere in the configuration, you can configure the MRRU by including the `mrru` statement at the `[edit interfaces lsq-`*`fpc/pic/port`*` unit `*`logical-unit-number`*`]` or `[edit interfaces `*`interface-name`*` mlfr-uni-nni-bundle-options]` hierarchy level. The MRRU is similar to the MTU, but is specific to link services interfaces. By default the MRRU size is 1500 bytes, and you can configure it to be from 1500 through 4500 bytes. For more information, see *Configuring MRRU on Multilink and Link Services Logical Interfaces*.

To configure fragmentation properties on a queue, include the `fragmentation-maps` statement at the `[edit class-of-service]` hierarchy level:

```
[edit class-of-service]
fragmentation-maps {
    map-name {
        forwarding-class class-name {
            (fragment-threshold bytes | no-fragmentation);
            multilink-class number;
        }
    }
}
```

To set a per-forwarding class fragmentation threshold, include the `fragment-threshold` statement in the fragmentation map. This statement sets the maximum size of each multilink fragment.

To set traffic on a queue to be nonencapsulated rather than multilink encapsulated, include the `no-fragmentation` statement in the fragmentation map. This statement specifies that an extra fragmentation header is not prepended to the packets received on this queue and that static link load balancing is used to ensure in-order packet delivery.

For a given forwarding class, you can include either the `fragment-threshold` or `no-fragmentation` statement; they are mutually exclusive.

You use the `multilink-class` statement to map a forwarding class into a multiclass MLPPP (MCML). For a given forwarding class, you can include either the `multilink-class` or `no-fragmentation` statement; they are mutually exclusive.

To associate a fragmentation map with a multilink PPP interface or MLFR FRF.16 DLCI, include the fragmentation-map statement at the [edit class-of-service interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit class-of-service interfaces]
lsq-fpc/pic/port {
    unit logical-unit-number { # Multilink PPP
    fragmentation-map map-name;
}
lsq-fpc/pic/port:channel { # MLFR FRF.16
unit logical-unit-number {
    fragmentation-map map-name;
}
```

For configuration examples, see the following topics:

- "Configuring LSQ Interfaces as NxT1 or NxE1 Bundles Using MLPPP" on page 939
- "Configuring LSQ Interfaces as NxT1 or NxE1 Bundles Using FRF.16" on page 939
- "Configuring LSQ Interfaces for Single Fractional T1 or E1 Interfaces Using MLPPP and LFI" on page 939
- "Configuring LSQ Interfaces for Single Fractional T1 or E1 Interfaces Using FRF.12" on page 939
- "Configuring LSQ Interfaces as NxT1 or NxE1 Bundles Using FRF.15" on page 939
- "Configuring LSQ Interfaces for T3 Links Configured for Compressed RTP over MLPPP" on page 939
- "Configuring LSQ Interfaces as T3 or OC3 Bundles Using FRF.12" on page 939
- "Configuring LSQ Interfaces for ATM2 IQ Interfaces Using MLPPP" on page 939

For Link Services PIC link services (ls-) interfaces, fragmentation maps are not supported. Instead, you enable LFI by including the interleave-fragments statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level. For more information, see *Configuring Delay-Sensitive Packet Interleaving on Link Services Logical Interfaces*.

SEE ALSO

Layer 2 Service Package Capabilities and Interfaces | **921**

Configuring Link Services and CoS on Services PICs

To configure link services and CoS on an AS or Multiservices PIC, you must perform the following steps:

1. Enable the Layer 2 service package. You enable service packages per PIC, not per port. When you enable the Layer 2 service package, the entire PIC uses the configured package. To enable the Layer 2 service package, include the `service-package` statement at the `[edit chassis fpc slot-number pic pic-number adaptive-services]` hierarchy level, and specify `layer-2`:

   ```
   [edit chassis fpc slot-number pic pic-number adaptive-services]
   service-package layer-2;
   ```

   For more information about AS or Multiservices PIC service packages, see *Enabling Service Packages* and "Layer 2 Service Package Capabilities and Interfaces" on page 921 .

2. Configure a multilink PPP or FRF.16 bundle by combining constituent links into a virtual link, or bundle.

   Configuring an MLPPP Bundle

   To configure an MLPPP bundle, configure constituent links and bundle properties by including the following statements in the configuration:

   ```
   [edit interfaces interface-name unit logical-unit-number]
   encapsulation ppp;
   family mlppp {
       bundle lsq-fpc/pic/port.logical-unit-number;
   }
   [edit interfaces lsq-fpc/pic/port unit logical-unit-number]
   drop-timeout milliseconds;
   encapsulation multilink-ppp;
   fragment-threshold bytes;
   link-layer-overhead percent;
   minimum-links number;
   mrru bytes;
   short-sequence;
   family inet {
       address address;
   }
   ```

For more information about these statements, see the Link and Multilink Services Interfaces User Guide for Routing Devices.

Configuring an MLFR FRF.16 Bundle

To configure an MLFR FRF.16 bundle, configure constituent links and bundle properties by including the following statements in the configuration:

```
[edit chassis fpc slot-number pic slot-number]
mlfr-uni-nni-bundles number;
[edit interfaces interface-name ]
encapsulation multilink-frame-relay-uni-nni;
unit logical-unit-number {
    family mlfr-uni-nni {
        bundle lsq-fpc/pic/port:channel;
    }
}
```

For more information about the `mlfr-uni-nni-bundles` statement, see the Junos OS Administration Library for Routing Devices. MLFR FRF.16 uses channels as logical units.

For MLFR FRF.16, you must configure one end as data circuit-terminating equipment (DCE) by including the following statements at the `[edit interfaces lsq-fpc/pic/port:channel]` hierarchy level.

```
encapsulation multilink-frame-relay-uni-nni;
dce;
mlfr-uni-nni-options {
    acknowledge-retries number;
    acknowledge-timer milliseconds;
    action-red-differential-delay (disable-tx | remove-link);
    drop-timeout milliseconds;
    fragment-threshold bytes;
    hello-timer milliseconds;
    link-layer-overhead percent;
    lmi-type (ansi | itu);
    minimum-links number;
    mrru bytes;
    n391 number;
    n392 number;
    n393 number;
```

```
        red-differential-delay milliseconds;
        t391 number;
        t392 number;
        yellow-differential-delay milliseconds;
    }
    unit logical-unit-number {
        dlci dlci-identifier;
        family inet {
            address address;
        }
    }
```

For more information about MLFR UNI NNI properties, see Link and Multilink Services Interfaces User Guide for Routing Devices.

3. To configure CoS components for each multilink bundle, enable per-unit scheduling on the interface, configure a scheduler map, apply the scheduler to each queue, configure a fragmentation map, and apply the fragmentation map to each bundle. Include the following statements:

```
[edit interfaces]
lsq-fpc/pic/port {
    per-unit-scheduler; # Enables per-unit scheduling on the bundle
}
[edit class-of-service]
interfaces {
    lsq-fpc/pic/port { # Multilink PPP
    unit logical-unit-number {
        scheduler-map map-name; # Applies scheduler map to each queue
    }
}
lsq-fpc/pic/port:channel { # MLFR FRF.16
unit logical-unit-number {
    # Scheduler map provides scheduling information for
    # the queues within a single DLCI.
    scheduler-map map-name;
    shaping-rate percent percent;
}
forwarding-classes {
    queue queue-number class-name priority (high | low);
}
scheduler-maps {
    map-name {
        forwarding-class class-name scheduler scheduler-name;
    }
}
schedulers {
    scheduler-name {
        buffer-size (percent percentage | remainder | temporal microseconds);
        priority priority-level;
        transmit-rate (percent percentage | rate | remainder) <exact>;
    }
}
fragmentation-maps {
    map-name {
        forwarding-class class-name {
            fragment-threshold bytes;
            no-fragmentation;
        }
    }
}
```

Associate a fragmentation map with a multilink PPP interface or MLFR FRF.16 DLCI by including the following statements at the `[edit class-of-service]` hierarchy level:

```
interfaces {
    lsq-fpc/pic/port {
        unit logical-unit-number { # Multilink PPP
        fragmentation-map map-name;
    }
}
lsq-fpc/pic/port:channel { # MLFR FRF.16
unit logical-unit-number {
    fragmentation-map map-name;
}
```

SEE ALSO

Layer 2 Service Package Capabilities and Interfaces | 921

Link Services Interface Redundancy | 921

Link Services Interface Redundancy | 921

Link Services Interface Redundancy | 921

Link Services Configuration for Junos Interfaces | 847

Oversubscribing Interface Bandwidth on LSQ Interfaces

IN THIS SECTION

- Examples: Oversubscribing an LSQ Interface | 864

The term *oversubscribing interface bandwidth* means configuring shaping rates (peak information rates [PIRs]) so that their sum exceeds the interface bandwidth.

On Channelized IQ PICs, Gigabit Ethernet IQ PICs, and FRF.16 link services IQ (`lsq-`) interfaces on AS and Multiservices PICs, you can oversubscribe interface bandwidth. The logical interfaces (and DLCIs within an FRF.16 bundle) can be oversubscribed when there is leftover bandwidth. The oversubscription is limited to the configured PIR. Any unused bandwidth is distributed equally among oversubscribed logical interfaces or DLCIs.

For networks that are not likely to experience congestion, oversubscribing interface bandwidth improves network utilization, thereby allowing more customers to be provisioned on a single interface. If the actual data traffic does not exceed the interface bandwidth, oversubscription allows you to sell more bandwidth than the interface can support.

We recommend avoiding oversubscription in networks that are likely to experience congestion. Be careful not to oversubscribe a service by too much, because this can cause degradation in the performance of the router during congestion. When you configure oversubscription, some output queues can be starved if the actual data traffic exceeds the physical interface bandwidth. You can prevent degradation by using statistical multiplexing to ensure that the actual data traffic does not exceed the interface bandwidth.

NOTE: You cannot oversubscribe interface bandwidth when you configure traffic shaping using the method described in *Applying Scheduler Maps and Shaping Rate to DLCIs and VLANs*.

When configuring oversubscription for FRF.16 bundle interfaces, you can assign traffic control profiles that apply on a physical interface basis. When you apply traffic control profiles to FRF.16 bundles at the *logical* interface level, member link interface bandwidth is underutilized when there is a small proportion of traffic or no traffic at all on an individual DLCI. Support for traffic control features on the FRF.16 bundle physical interface level addresses this limitation.

To configure oversubscription of an interface, perform the following steps:

1. Include the `shaping-rate` statement at the `[edit class-of-service traffic-control-profiles profile-name]` hierarchy level:

    ```
    [edit class-of-service traffic-control-profiles profile-name]
    shaping-rate (percent percentage | rate);
    ```

    > **NOTE**: When configuring oversubscription for FRF.16 bundle interfaces on a physical interface basis, you *must* specify `shaping-rate` as a percentage.

    On LSQ interfaces, you can configure the shaping rate as a percentage.

    On IQ and IQ2 interfaces, you can configure the shaping rate as an absolute rate from 1000 through 6,400,000,000,000 bits per second.

    Alternatively, you can configure a shaping rate for a logical interface and oversubscribe the physical interface by including the `shaping-rate` statement at the `[edit class-of-service interfaces interface-name unit logical-unit-number]` hierarchy level. However, with this configuration approach, you cannot independently control the delay-buffer rate, as described in Step 2.

    > **NOTE**: For channelized and Gigabit Ethernet IQ interfaces, the `shaping-rate` and `guaranteed-rate` statements are mutually exclusive. You cannot configure some logical interfaces to use a shaping rate and others to use a guaranteed rate. This means there are no service guarantees when you configure a PIR. For these interfaces, you can configure either a PIR or a committed information rate (CIR), but not both.
    >
    > This restriction does not apply to Gigabit Ethernet IQ2 PICs or link services IQ (LSQ) interfaces on AS or Multiservices PICs. For LSQ and Gigabit Ethernet IQ2 interfaces, you can configure both a PIR and a CIR on an interface. For more information about CIRs, see "Configuring Guaranteed Minimum Rate on LSQ Interfaces" on page 866 .

2. Optionally, you can base the delay buffer calculation on a delay-buffer rate. To do this, include the `delay-buffer-rate` statement at the `[edit class-of-service traffic-control-profiles profile-name]` hierarchy level:

NOTE: When configuring oversubscription for FRF.16 bundle interfaces on a physical interface basis, you *must* specify `delay-buffer-rate` as a percentage.

```
[edit class-of-service traffic-control-profiles profile-name]
delay-buffer-rate (percent percentage | rate);
```

The delay-buffer rate overrides the shaping rate as the basis for the delay-buffer calculation. In other words, the shaping rate or scaled shaping rate is used for delay-buffer calculations only when the delay-buffer rate is not configured.

For LSQ interfaces, if you do not configure a delay-buffer rate, the guaranteed rate (CIR) is used to assign buffers. If you do not configure a guaranteed rate, the shaping rate (PIR) is used in the undersubscribed case, and the scaled shaping rate is used in the oversubscribed case.

On LSQ interfaces, you can configure the delay-buffer rate as a percentage.

On IQ and IQ2 interfaces, you can configure the delay-buffer rate as an absolute rate from 1000 through 6,400,000,000,000 bits per second.

The actual delay buffer is based on the calculations described in the Class of Service User Guide (Routers and EX9200 Switches). For an example showing how the delay-buffer rates are applied, see "Examples: Oversubscribing an LSQ Interface" on page 864 .

Configuring large buffers on relatively low-speed links can cause packet aging. To help prevent this problem, the software requires that the sum of the delay-buffer rates be less than or equal to the port speed.

This restriction does not eliminate the possibility of packet aging, so you should be cautious when using the `delay-buffer-rate` statement. Though some amount of extra buffering might be desirable for burst absorption, delay-buffer rates should not far exceed the service rate of the logical interface.

If you configure delay-buffer rates so that the sum exceeds the port speed, the configured delay-buffer rate is not implemented for the last logical interface that you configure. Instead, that logical interface receives a delay-buffer rate of zero, and a warning message is displayed in the CLI. If bandwidth becomes available (because another logical interface is deleted or deactivated, or the port speed is increased), the configured delay-buffer-rate is reevaluated and implemented if possible.

If you do not configure a delay-buffer rate or a guaranteed rate, the logical interface receives a delay-buffer rate in proportion to the shaping rate and the remaining delay-buffer rate available. In other words, the delay-buffer rate for each logical interface with no configured delay-buffer rate is equal to:

```
(remaining delay-buffer rate * shaping rate) / (sum of shaping rates)
```

The remaining delay-buffer rate is equal to:

```
(interface speed) - (sum of configured delay-buffer rates)
```

3. To assign a scheduler map to the logical interface, include the `scheduler-map` statement at the `[edit class-of-service traffic-control-profiles profile-name]` hierarchy level:

   ```
   [edit class-of-service traffic-control-profiles profile-name]
   scheduler-map map-name;
   ```

   For information about configuring schedulers and scheduler maps, see the Class of Service User Guide (Routers and EX9200 Switches).

4. Optionally, you can enable large buffer sizes to be configured. To do this, include the `q-pic-large-buffer` statement at the `[edit chassis fpc slot-number pic pic-number]` hierarchy level:

   ```
   [edit chassis fpc slot-number pic pic-number]
   q-pic-large-buffer;
   ```

   If you do not include this statement, the delay-buffer size is more restricted. We recommend restricted buffers for delay-sensitive traffic, such as voice traffic. For more information, see the Class of Service User Guide (Routers and EX9200 Switches).

5. To enable scheduling on logical interfaces, include the `per-unit-scheduler` statement at the `[edit interfaces interface-name]` hierarchy level:

   ```
   [edit interfaces interface-name ]
   per-unit-scheduler;
   ```

   When you include this statement, the maximum number of VLANs supported is 768 on a single-port Gigabit Ethernet IQ PIC. On a two-port Gigabit Ethernet IQ PIC, the maximum number is 384.

6. To enable scheduling for FRF.16 bundles physical interfaces, include the no-per-unit-scheduler statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
no-per-unit-scheduler;
```

7. To apply the traffic-scheduling profile to the logical interface, include the output-traffic-control-profile statement at the [edit class-of-service interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit class-of-service interfaces interface-name unit logical-unit-number]
output-traffic-control-profile profile-name;
```

   You cannot include the output-traffic-control-profile statement in the configuration if any of the following statements are included in the logical interface configuration: scheduler-map, shaping-rate, adaptive-shaper, or virtual-channel-group.

   For a table that shows how the bandwidth and delay buffer are allocated in various configurations, see the Class of Service User Guide (Routers and EX9200 Switches).

Examples: Oversubscribing an LSQ Interface

Oversubscribing an LSQ Interface with Scheduling Based on the Logical Interface

Apply a traffic-control profile to a logical interface representing a DLCI on an FRF.16 bundle.

```
interfaces {
    lsq-1/3/0:0 {
        per-unit-scheduler;
        unit 0 {
            dlci 100;
        }
        unit 1 {
            dlci 200;
        }
    }
}
class-of-service {
    traffic-control-profiles {
        tc_0 {
            shaping-rate percent 100;
            guaranteed-rate percent 60;
            delay-buffer-rate percent 80;
        }
        tc_1 {
            shaping-rate percent 80;
            guaranteed-rate percent 40;
        }
    }
    interfaces {
        lsq-1/3/0 {
            unit 0 {
                output-traffic-control-profile tc_0;
            }
            unit 1 {
                output-traffic-control-profile tc_1;
            }
        }
    }
}
```

Oversubscribing an LSQ Interface with Scheduling Based on the Physical Interface

Apply a traffic-control profile to the physical interface representing an FRF.16 bundle:

```
interfaces {
    lsq-0/2/0:0 {
        no-per-unit-scheduler;
        encapsulation multilink-frame-relay-uni-nni;
        unit 0 {
            dlci 100;
            family inet {
                address 18.18.18.2/24;
            }
        }
    }
class-of-service {
    traffic-control-profiles {
        rlsq_tc {
            scheduler-map rlsq;
            shaping-rate percent 60;
            delay-buffer-rate percent 10;
        }
    }
    interfaces {
        lsq-0/2/0:0 {
            output-traffic-control-profile rlsq_tc;
        }
    }
}
scheduler-maps {
    rlsq {
        forwarding-class best-effort scheduler rlsq_scheduler;
        forwarding-class expedited-forwarding scheduler rlsq_scheduler1;
    }
}
schedulers {
    rlsq_scheduler {
        transmit-rate percent 20;
        priority low;
    }
    rlsq_scheduler1 {
        transmit-rate percent 40;
        priority high;
    }
}
```

SEE ALSO

Layer 2 Service Package Capabilities and Interfaces | 921

Inline Multlink Services | 939

Link Services Configuration for Junos Interfaces | 847

Configuring Guaranteed Minimum Rate on LSQ Interfaces

IN THIS SECTION

- Example: Configuring Guaranteed Minimum Rate | 869

On Gigabit Ethernet IQ PICs, Channelized IQ PICs, and FRF.16 link services IQ (LSQ) interfaces on AS and Multiservices PICs, you can configure guaranteed bandwidth, also known as a committed information rate (CIR). This allows you to specify a guaranteed rate for each logical interface. The guaranteed rate is a minimum. If excess physical interface bandwidth is available for use, the logical interface receives more than the guaranteed rate provisioned for the interface.

You cannot provision the sum of the guaranteed rates to be more than the physical interface bandwidth, or the bundle bandwidth for LSQ interfaces. If the sum of the guaranteed rates exceeds the interface or bundle bandwidth, the commit operation does not fail, but the software automatically decreases the rates so that the sum of the guaranteed rates is equal to the available bundle bandwidth.

To configure a guaranteed minimum rate, perform the following steps:

1. Include the `guaranteed-rate` statement at the `[edit class-of-service traffic-control-profiles` *`profile-name`*`]` hierarchy level:

   ```
   [edit class-of-service traffic-control-profiles profile-name]
   guaranteed-rate (percent percentage | rate);
   ```

   On LSQ interfaces, you can configure the guaranteed rate as a percentage.

   On IQ and IQ2 interfaces, you can configure the guaranteed rate as an absolute rate from 1000 through 160,000,000,000 bits per second.

   NOTE: For channelized and Gigabit Ethernet IQ interfaces, the `shaping-rate` and `guaranteed-rate` statements are mutually exclusive. You cannot configure some logical interfaces to use a shaping rate and others to use a guaranteed rate. This means there are no service guarantees when you configure a PIR. For these interfaces, you can configure either a PIR or a committed information rate (CIR), but not both.

   This restriction does not apply to Gigabit Ethernet IQ2 PICs or link services IQ (LSQ) interfaces on AS or Multiservices PICs. For LSQ and Gigabit Ethernet IQ2 interfaces, you can configure both a PIR and a CIR on an interface. For more information about CIRs, see the Class of Service User Guide (Routers and EX9200 Switches).

2. Optionally, you can base the delay buffer calculation on a delay-buffer rate. To do this, include the `delay-buffer-rate` statement at the `[edit class-of-service traffic-control-profiles` *`profile-name`*`]` hierarchy level:

   ```
   [edit class-of-service traffic-control-profiles profile-name]
   delay-buffer-rate (percent percentage | rate);
   ```

   On LSQ interfaces, you can configure the delay-buffer rate as a percentage.

   On IQ and IQ2 interfaces, you can configure the delay-buffer rate as an absolute rate from 1000 through 160,000,000,000 bits per second.

The actual delay buffer is based on the calculations described in tables in the Class of Service User Guide (Routers and EX9200 Switches). For an example showing how the delay-buffer rates are applied, see "Example: Configuring Guaranteed Minimum Rate" on page 869 .

If you do not include the `delay-buffer-rate` statement, the delay-buffer calculation is based on the guaranteed rate, the shaping rate if no guaranteed rate is configured, or the scaled shaping rate if the interface is oversubscribed.

If you do not specify a shaping rate or a guaranteed rate, the logical interface receives a minimal delay-buffer rate and minimal bandwidth equal to 4 MTU-sized packets.

You can configure a rate for the delay buffer that is higher than the guaranteed rate. This can be useful when the traffic flow might not require much bandwidth in general, but in some cases can be bursty and therefore needs a large buffer.

Configuring large buffers on relatively low-speed links can cause packet aging. To help prevent this problem, the software requires that the sum of the delay-buffer rates be less than or equal to the port speed. This restriction does not eliminate the possibility of packet aging, so you should be cautious when using the `delay-buffer-rate` statement. Though some amount of extra buffering might be desirable for burst absorption, delay-buffer rates should not far exceed the service rate of the logical interface.

If you configure delay-buffer rates so that the sum exceeds the port speed, the configured delay-buffer rate is not implemented for the last logical interface that you configure. Instead, that logical interface receives a delay-buffer rate of 0, and a warning message is displayed in the CLI. If bandwidth becomes available (because another logical interface is deleted or deactivated, or the port speed is increased), the configured delay-buffer-rate is reevaluated and implemented if possible.

If the guaranteed rate of a logical interface cannot be implemented, that logical interface receives a delay-buffer rate of 0, even if the configured delay-buffer rate is within the interface speed. If at a later time the guaranteed rate of the logical interface can be met, the configured delay-buffer rate is reevaluated and if the delay-buffer rate is within the remaining bandwidth, it is implemented.

If any logical interface has a configured guaranteed rate, all other logical interfaces on that port that do not have a guaranteed rate configured receive a delay-buffer rate of 0. This is because the absence of a guaranteed rate configuration corresponds to a guaranteed rate of 0 and, consequently, a delay-buffer rate of 0.

3. To assign a scheduler map to the logical interface, include the `scheduler-map` statement at the `[edit class-of-service traffic-control-profiles *profile-name*]` hierarchy level:

```
[edit class-of-service traffic-control-profiles profile-name]
scheduler-map map-name;
```

For information about configuring schedulers and scheduler maps, see the Class of Service User Guide (Routers and EX9200 Switches).

4. To enable large buffer sizes to be configured, include the `q-pic-large-buffer` statement at the `[edit chassis fpc slot-number pic pic-number]` hierarchy level:

   ```
   [edit chassis fpc slot-number pic pic-number]
   q-pic-large-buffer;
   ```

   If you do not include this statement, the delay-buffer size is more restricted. For more information, see the Class of Service User Guide (Routers and EX9200 Switches).

5. To enable scheduling on logical interfaces, include the `per-unit-scheduler` statement at the `[edit interfaces interface-name]` hierarchy level:

   ```
   [edit interfaces interface-name ]
   per-unit-scheduler;
   ```

   When you include this statement, the maximum number of VLANs supported is 767 on a single-port Gigabit Ethernet IQ PIC. On a two-port Gigabit Ethernet IQ PIC, the maximum number is 383.

6. To apply the traffic-scheduling profile to the logical interface, include the output-traffic-control-profile statement at the `[edit class-of-service interfaces interface-name unit logical-unit-number]` hierarchy level:

   ```
   [edit class-of-service interfaces interface-name unit logical-unit-number]
   output-traffic-control-profile profile-name;
   ```

Example: Configuring Guaranteed Minimum Rate

Two logical interface units, `0` and `1`, are provisioned with a guaranteed minimum of 750 Kbps and 500 Kbps, respectively. For logical unit `1`, the delay buffer is based on the guaranteed rate setting. For logical unit `0`, a delay-buffer rate of 500 Kbps is specified. The actual delay buffers allocated to each logical interface are 2 seconds of 500 Kbps. The 2-second value is based on the following calculation:

```
delay-buffer-rate < [8 x 64 Kbps]): 2 seconds of delay-buffer-rate
```

For more information about this calculation, see the Class of Service User Guide (Routers and EX9200 Switches).

```
chassis {
    fpc 3 {
        pic 0 {
            q-pic-large-buffer;
        }
    }
}
interfaces {
    t1-3/0/1 {
        per-unit-scheduler;
    }
}
class-of-service {
    traffic-control-profiles {
        tc-profile3 {
            guaranteed-rate 750k;
            scheduler-map sched-map3;
            delay-buffer-rate 500k; # 500 Kbps is less than 8 x 64 Kbps
        }
        tc-profile4 {
            guaranteed-rate 500k; # 500 Kbps is less than 8 x 64 Kbps
            scheduler-map sched-map4;
        }
    }
    interface t1-3/0/1 {
        unit 0 {
            output-traffic-control-profile tc-profile3;
        }
        unit 1 {
            output-traffic-control-profile tc-profile4;
        }
    }
}
```

SEE ALSO

Layer 2 Service Package Capabilities and Interfaces | 921

Inline Multlink Services | 939

Link Services Configuration for Junos Interfaces | 847

9

PART

Inter-Chassis Redundancy for NAT and Stateful Firewall Flows

Configuring Inter-Chassis MS-MPC and MS-MIC for NAT and Stateful Firewall (Release 16.1 and later) | 873

Configuring Inter-Chassis Stateful Synchronization for NAT and Stateful Firewall (Release 15.1 and earlier) | 903

CHAPTER 34

Configuring Inter-Chassis MS-MPC and MS-MIC for NAT and Stateful Firewall (Release 16.1 and later)

IN THIS CHAPTER

- Inter-Chassis Stateful Synchronization for Long Lived NAT and Stateful Firewall Flows (MS-MPC, MS-MIC) (Release 16.1 and later) | 873
- Service Redundancy Daemon | 890

Inter-Chassis Stateful Synchronization for Long Lived NAT and Stateful Firewall Flows (MS-MPC, MS-MIC) (Release 16.1 and later)

IN THIS SECTION

- Configuring Inter-chassis MS-MPC and MS-MIC Redundancy for NAT and Stateful Firewall Overview (Release 16.1 and later) | 874
- Inter-Chassis Stateful Synchronization for Long Lived NAT and Stateful Firewall Flows (MS-MPC, MS-MIC) Overview (Release 16.1 and later) | 874
- Configuring Inter-Chassis Stateful Synchronization for Long Lived NAT and Stateful Firewall Flows (MS-MPC, MS-MIC) (Release 16.1 and later) | 876
- Example: Inter-Chassis Stateful Synchronization for Long-Lived NAT and Stateful Firewall Flows (MS-MIC, MS-MPC) (Release 16.1 and later) | 878

Configuring Inter-chassis MS-MPC and MS-MIC Redundancy for NAT and Stateful Firewall Overview (Release 16.1 and later)

NOTE: This topic applies to Junos OS release 16.1 and higher. (For Junos OS release 15.1 and earlier, see "Inter-Chassis High Availability for MS-MIC and MS-MPC (Release 15.1 and earlier)" on page 903).

Carrier-grade NAT (CGN) and stateful firewall deployments can use a dual-chassis implementation to provide a redundant data path and redundancy for key components in the router. Although intra-chassis high availability can be used in an MX Series device by employing the AMS interfaces, this method only deals locally with service PIC and full MS-MPC or MS-MIC card failures. If for any reason traffic is switched to a backup router due to some other failure in the router, the session state from the Service PICs is lost. Inter-chassis high availability offers a more robust solution by preserving the session state of NAT and stateful firewalls from the services PICs. This technology is a primary-secondary model, not an active-active cluster. Traffic to be serviced by the services PICs that are configured for inter-chassis high availability only flows through the MX Series device that is currently the primary in the pair.

To configure interchassis redundancy for NAT and stateful firewall, you configure:

1. Stateful synchronization, which replicates the session state from the services PICs on the primary chassis to the backup chassis. For more information, see "Inter-Chassis Stateful Synchronization for Long Lived NAT and Stateful Firewall Flows (MS-MPC, MS-MIC) Overview (Release 16.1 and later)" on page 874 .

2. The service redundancy daemon, which allows primary-role switchover to occur based on a monitored event. Most operators would not want to employ stateful synchronization without also implementing the service redundancy daemon. For more information, see "Service Redundancy Daemon Overview" on page 891

Inter-Chassis Stateful Synchronization for Long Lived NAT and Stateful Firewall Flows (MS-MPC, MS-MIC) Overview (Release 16.1 and later)

NOTE: This topic applies to Junos OS release 16.1 and higher. (For Junos OS release 15.1 and earlier, see "Inter-Chassis High Availability for MS-MIC and MS-MPC (Release 15.1 and earlier)" on page 903).

Stateful synchronization synchronizes long-lived sessions between the primary and backup MX Series chassis in the high availability pair. By default, long lived sessions are stateful firewall, NAT, and IDS sessions that have been active on the services PIC for 180 seconds, though you can configure this to be a higher or lower value. Stateful firewall sessions, NAT sessions, and IDS sessions are the session types that can be synchronized.

Inter-chassis high availability works with ms- service interfaces configured on MS-MIC or MS-MPC interface cards. An ms- interface unit other than unit 0 must be configured with the `ip-address-owner service-plane` option.

The following NAT translation types and sessions support stateful synchronization:

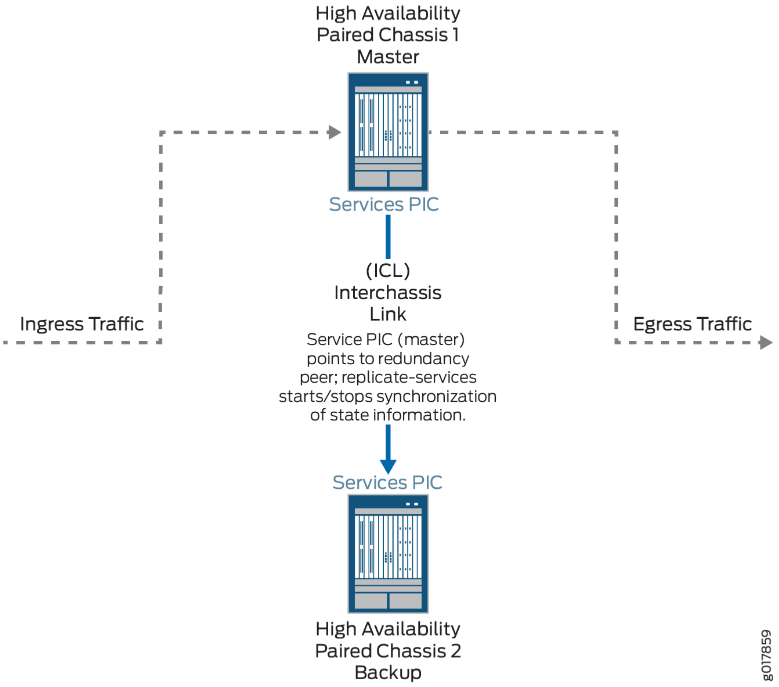
- basic-nat44
- dynamic-nat44
- napt-44
- napt-44 with endpoint-independent mapping (EIM), or endpoint-independent filters (EIF)
- dnat-44
- twice-nat
- stateful-nat64

The following restrictions apply:

- Replicating state information for the port block allocation (PBA), endpoint-independent mapping (EIM), or endpoint-independent filters (EIF) features is not supported.
- When configuring a service set for NAT or stateful firewall that belongs to a stateful synchronization setup, - the NAT and stateful firewall configurations for the service set must be identical on both MX Series devices.
- Application Layer Gateway (ALG) sessions do not support stateful synchronization.

Figure 43 on page 876 shows the inter-chassis high availability topology.

Figure 43: Stateful Sync Topology



Configuring Inter-Chassis Stateful Synchronization for Long Lived NAT and Stateful Firewall Flows (MS-MPC, MS-MIC) (Release 16.1 and later)

NOTE: This topic applies to Junos OS release 16.1 and higher. (For Junos OS release 15.1 and earlier, see "Inter-Chassis High Availability for MS-MIC and MS-MPC (Release 15.1 and earlier)" on page 903).

To configure stateful synchronization inter-chassis high availability for stateful firewall and NAPT44 on MS-MIC or MS-MPC service PICs, perform the following configuration steps on each chassis of the high availability pair.

1. Configure the services ms- interface.

   a. Specify the IPv4 address of the local services card. This address is used by the TCP channel between the HA pairs.

   ```
   [edit interfaces interface-name redundancy-options]
   user@host# set redundancy-local data-address address
   ```

   When you configure the other chassis, this is the address you use for the `redundancy-peer ipaddress`.

b. Specify the IPv4 address of the remote services card. This address is used by the TCP channel between the HA pairs.

```
[edit interfaces interface-name redundancy-options]
user@host# set redundancy-peer ipaddress address
```

When you configure the other chassis, this is the address you use for the `redundancy-local data-address`.

c. Configure the length of time that the flow remains active for replication, in seconds.

```
[edit interfaces interface-name redundancy-options]
user@host# set replication-threshold seconds
```

d. Configure a unit other than 0 with the `ip-address-owner service-plane` option.

```
[edit interfaces interface-name]
user@host# set unit logical-unit-number ip-address-owner service-plane
```

e. For the unit configured with the `ip-address-owner service-plane` option, assign the IPv4 address of the local services card that you configured with the `redundancy-local data-address` option.

```
[edit interfaces interface-name unit logical-unit-number]
user@host# set family inet addressaddress
```

f. Configure the inside and outside interface units, which are used by the next-hop service set. Use different unit numbers for the inside and outside units, and do not use 0 or the unit number used with the `ip-address-owner service-plane` option.

```
[edit]
user@host# set interfaces interface-name unit logical-unit-number family inet
user@host# set interfaces interface-name unit logical-unit-number service-domain inside
user@host# set interfaces interface-name unit logical-unit-number family inet
user@host# set interfaces interface-name unit logical-unit-number service-domain outside
```

2. Configure the next-hop service set that contains the NAT rules or stateful firewall rules. The service set must be configured identically on each chassis of the high availability pair. The NAT rules and stateful firewall rules must also be configured identically on each chassis.

3. For ease of management, we recommend you create a special routing instance with `instance-type vrf` to host the HA synchronization traffic between the MX Series high availability pair. Then specify the name of the special routing instance to apply to the HA synchronization traffic between the high availability pair.

```
[edit interfaces interface-name redundancy-options]
user@host# set routing-instance instance-name
```

4. Repeat these steps for the other chassis of the high availability pair.

Example: Inter-Chassis Stateful Synchronization for Long-Lived NAT and Stateful Firewall Flows (MS-MIC, MS-MPC) (Release 16.1 and later)

IN THIS SECTION

- Requirements | 878
- Overview | 878
- Configuration | 879

This example shows how to configure inter-chassis high availability for NAT services.

Requirements

This example uses the following hardware and software components:

- Two MX480 routers with MS-MPC line cards
- Junos OS Release 16.1 or later

Overview

Two MX Series routers are identically configured to facilitate stateful failover for NAT services in case of a chassis failure.

Configuration

IN THIS SECTION

- CLI Quick Configuration | 879
- Configuring Interfaces for Chassis 1 | 881
- Configure Routing Information for HA Synchronization Traffic Between MX Series Routers for Chassis 1 | 883
- Configuring NAT for Chassis 1 | 884
- Configuring the Service Set | 886
- Configuring Interfaces for Chassis 2 | 887
- Configure Routing Information for HA Synchronization Traffic Between MX Series Routers for Chassis 2 | 889

To configure inter-chassis high availability for this example, perform these tasks:

CLI Quick Configuration

To quickly configure this example on the routers, copy the following commands and paste them into the router terminal window after removing line breaks and substituting interface information specific to your site.

NOTE: The following configuration is for chassis 1.

```
[edit]
set interfaces ms-4/0/0 redundancy-options redundancy-peer ipaddress 5.5.5.2
set interfaces ms-4/0/0 redundancy-options redundancy-local data-address 5.5.5.1
set interfaces ms-4/0/0 redundancy-options routing-instance HA
set interfaces ms-4/0/0 redundancy-options replication-threshold 180
set interfaces ms-4/0/0 unit 10 ip-address-owner service-plane
set interfaces ms-4/0/0 unit 10 family inet address 5.5.5.1/32
set interfaces ms-4/0/0 unit 20 family inet
set interfaces ms-4/0/0 unit 20 service-domain inside
set interfaces ms-4/0/0 unit 30 family inet
set interfaces ms-4/0/0 unit 30 service-domain outside
set interfaces ge-2/0/0 vlan-tagging
set interfaces ge-2/0/0 unit 0 vlan-id 100 family inet address 20.1.1.1/24
set policy-options policy-statement dummy term 1 then reject
set routing-instances HA instance-type vrf
set routing-instances HA interface ge-2/0/0.0
set routing-instances HA interface ms-4/0/0.10
set routing-instances HA route-distinguisher 1:1
set routing-instances HA vrf-import dummy
set routing-instances HA vrf-export dummy
set routing-instances HA routing-options static route route 5.5.5.1/32 next-hop ms-4/0/0.10
set routing-instances HA routing-options static route route 5.5.5.2/32 next-hop 20.1.1.2
set routing-options static-route 100.100.100.0/24 next-hop ms-4/0/0.20
set services nat pool p2 address 32.0.0.0/24
set services nat pool p2 port automatic random-allocation
set services nat pool p2 address-allocation round-robin
set services nat rule r2 match-direction input
set services nat rule r2 term t1 from source-address 129.0.0.0/8
set services nat rule r2 term t1 from source-address 128.0.0.0/8
set services nat rule r2 term t1 then translated source-pool p2
set services nat rule r2 term t1 then translated translation-type napt-44
set services nat rule r2 term t1 then translated address-pooling paired
set services nat rule r2 term t1 then syslog
set services service-set ss2 nat-rules r2
set services service-set ss2 next-hop-service inside-service-interface ms-4/0/0.20
set services service-set ss2 next-hop-service outside-service-interface ms-4/0/0.30
set services service-set ss2 syslog host local class session-logs
set services service-set ss2 syslog host local class nat-logs
```

NOTE: The following configuration is for chassis 2. NAT and service set information must be identical for chassis 1 and 2.

```
set interfaces ms-4/0/0 redundancy-options redundancy-peer ipaddress 5.5.5.1
set interfaces ms-4/0/0 redundancy-options redundancy-local data-address 5.5.5.2
set interfaces ms-4/0/0 redundancy-options routing-instance HA
set interfaces ms-4/0/0 redundancy-options replication-threshold 180
set interfaces ms-4/0/0 unit 10 ip-address-owner service-plane
set interfaces ms-4/0/0 unit 10 family inet address 5.5.5.2/32
set interfaces ms-4/0/0 unit 20 family inet
set interfaces ms-4/0/0 unit 20 service-domain inside
set interfaces ms-4/0/0 unit 30 family inet
set interfaces ms-4/0/0 unit 30 service-domain outside
set interfaces ge-2/0/0 vlan-tagging
set interfaces ge-2/0/0 unit 0 vlan-id 100 family inet address 20.1.1.2/24
set policy-options policy-statement dummy term 1 then reject
set routing-instances HA instance-type vrf
set routing-instances HA interface ge-2/0/0.0
set routing-instances HA interface ms-4/0/0.10
set routing-instances HA route-distinguisher 1:1
set routing-instances HA vrf-import dummy
set routing-instances HA vrf-export dummy
set routing-instances HA routing-options static route 5.5.5.2/32 next-hop ms-4/0/0.10
set routing-instances HA routing-options static route 5.5.5.1/32 next-hop 20.1.1.1
set routing-options static-route 100.100.100.0/24 next-hop ms-4/0/0.20
set services nat pool p2 address 32.0.0.0/24
set services nat pool p2 port automatic random-allocation
set services nat pool p2 address-allocation round-robin
set services nat rule r2 match-direction input
set services nat rule r2 term t1 from source-address 129.0.0.0/8
set services nat rule r2 term t1 from source-address 128.0.0.0/8
set services nat rule r2 term t1 then translated source-pool p2
set services nat rule r2 term t1 then translated translation-type napt-44
set services nat rule r2 term t1 then translated address-pooling paired
set services nat rule r2 term t1 then syslog
set services service-set ss2 nat-rules r2
set services service-set ss2 next-hop-service inside-service-interface ms-4/0/0.20
set services service-set ss2 next-hop-service outside-service-interface ms-4/0/0.30
set services service-set ss2 syslog host local class session-logs
set services service-set ss2 syslog host local class nat-logs
```

Configuring Interfaces for Chassis 1

Step-by-Step Procedure

The interfaces for each of the HA pair of routers are configured identically with the exception of the following service PIC options:

- The `redundancy-options redundancy-peer ipaddress` *`address`* must be different on each chassis *and* must point to the `redundancy-options redundancy-local data-address` *`data-address`* on the peer chassis.

- The `unit` *`unit-number`* `family inet address` *`address`* of a unit, other than 0, that contains the `ip-address-owner service-plane` option must be different on each chassis.

To configure interfaces:

1. Configure the redundant service PIC on chassis 1.

```
[edit interfaces}
user@host# set interfaces ms-4/0/0 redundancy-options redundancy-peer ipaddress 5.5.5.2
user@host# set interfaces ms-4/0/0 redundancy-options redundancy-local data-address 5.5.5.1
user@host# set interfaces ms-4/0/0 redundancy-options routing-instance HA
user@host# set interfaces ms-4/0/0 redundancy-options replication-threshold 180
user@host# set interfaces ms-4/0/0 unit 10 ip-address-owner service-plane
user@host# set interfaces ms-4/0/0 unit 10 family inet address 5.5.5.1/32
user@host# set interfaces ms-4/0/0 unit 20 family inet
user@host# set interfaces ms-4/0/0 unit 20 service-domain inside
user@host# set interfaces ms-4/0/0 unit 30 family inet
user@host# set interfaces ms-4/0/0 unit 30 service-domain outside
```

2. Configure the interfaces for chassis 1 that are used as interchassis links for synchronization traffic.

```
user@host# set interfaces ge-2/0/0 vlan-tagging
user@host# set interfaces ge-2/0/0 unit 0 vlan-id 100 family inet address 20.1.1.1/24
```

3. Configure remaining interfaces as needed.

Results

```
user@host# show interfaces

ge-2/0/0 {
    vlan-tagging;
    unit 0 {
        vlan-id 100;
        family inet {
             address 20.1.1.1/24;
        }
    }
}
ms-4/0/0 {
    redundancy-options {
        redundancy-peer {
            address 5.5.5.2;
        }
        redundancy-local {
            data-address 5.5.5.1;
        }
        routing-instance HA;
    }
    unit 10 {
        ip-address-owner service-plane;
        family inet {
            address 5.5.5.1/32;
        }
    }
    unit 20 {
        family inet;
        family inet6;
        service-domain inside;
    }
    unit 30 {
        family inet;
        family inet6;
        service-domain outside;
    }
}
```

Configure Routing Information for HA Synchronization Traffic Between MX Series Routers for Chassis 1

Step-by-Step Procedure

Detailed routing configuration is not included for this example. A routing instance is required for the HA synchronization traffic between the chassis as follows:

To configure the routing instances for chassis 1:

1. Specify a dummy policy statement. This statement is referenced in the routing instance configuration.

   ```
   user@host# set policy-options policy-statement dummy term 1 then reject
   ```

2. Specify the options for the routing instance.

   ```
   user@host# set routing-instances HA instance-type vrf
   user@host# set routing-instances HA interface ge-2/0/0.0
   ```

```
user@host# set routing-instances HA interface ms-4/0/0.10
user@host# set routing-instances HA route-distinguisher 1:1
user@host# set policy-options policy-statement dummy term 1 then reject
user@host# set routing-instances HA vrf-import dummy
user@host# set routing-instances HA vrf-export dummy
@user@host# set routing-instances HA routing-options static route 5.5.5.1/32 next-hop
ms-4/0/0.10
user@host# set routing-instances HA routing-options static route 5.5.5.2/32 next-hop 20.1.1.2
```

3. Specify the next-hop traffic to which the service set is applied.

```
user@host# set routing-options static-route 100.100.100.0/24 next-hop ms-4/0/0.20
```

Results

```
@user@host# show routing-instances
HA {
    instance-type vrf;
    interface ge-2/0/0.0;
    interface ms-4/0/0.10;
    route-distinguisher 1:1;
    vrf-import dummy;
    vrf-export dummy;
    routing-options {
        static {
            route 5.5.5.1/32 next-hop ms-4/0/0.10;
            route 5.5.5.2/32 next-hop 20.1.1.2;
        }
    }
}
```

Configuring NAT for Chassis 1

Step-by-Step Procedure

Configure NAT identically on both routers.

To configure NAT:

1. Specify NAT pool and rule information..

```
user@host# set services nat pool p2 address 32.0.0.0/24
user@host# set services nat pool p2 port automatic random-allocation
user@host# set services nat pool p2 address-allocation round-robin
user@host# set services nat rule r2 match-direction input
user@host# set services nat rule r2 term t1 from source-address 129.0.0.0/8
user@host# set services nat rule r2 term t1 from source-address 128.0.0.0/8
user@host# set services nat rule r2 term t1 then translated source-pool p2
user@host# set services nat rule r2 term t1 then translated translation-type napt-44
user@host# set services nat rule r2 term t1 then translated address-pooling paired
user@host# set services nat rule r2 term t1 then syslog
```

Results

```
user@host# show services nat
nat {
    pool p2 {
        address 32.0.0.0/24;
        port {
            automatic {
                random-allocation;
            }
        }
        address-allocation round-robin;
    }
    rule r2 {
        match-direction input;
        term t1 {
            from {
                source-address {
                    129.0.0.0/8;
                    128.0.0.0/8;
                }
            }
            then {
                translated {
                    source-pool p2;
                    translation-type {
                        napt-44;
                    }
                    address-pooling paired;
                }
                syslog;
            }
        }
    }
}
```

Configuring the Service Set

Step-by-Step Procedure

Configure the service set identically on both routers. To configure the service set:

1. (Optional) Service sets are replicated by default. To exclude a service set from replication using the following option.

   ```
   user@host# set services service-set ss2 replicate-services  disable-replication-capability
   ```

2. Configure references to NAT rules for the service set.

   ```
   user@host# set services service-set ss2 nat-rules r2
   ```

3. Configure next-hop service interface on the MS-PIC.

   ```
   user@host# set services service-set ss2 next-hop-service inside-service-interface ms-4/0/0.20
   user@host# set services service-set ss2 next-hop-service outside-service-interface ms-4/0/0.30
   ```

4. Configure desired logging options.

   ```
   user@host# set services service-set ss2 syslog host local class session-logs
   user@host# set services service-set ss2 syslog host local class nat-logs
   ```

Results

```
user@host# show services service-set ss2
syslog {
   host local {
        class {
            session-logs;
            inactive:
            nat-logs;
        }
    }
    replicate-services {
        replication-threshold 180;
        inactive: disable-replication-capability;
    }
    nat-rules r2;
    next-hop-service {
        inside-service-interface ms-3/0/0.20;
        outside-service-interface ms-3/0/0.30;
    }
}
```

Configuring Interfaces for Chassis 2

Step-by-Step Procedure

The interfaces for each of the HA pair of routers are configured identically with the exception of the following service PIC options:

- `redundancy-options redundancy-peer ipaddress` *address*
- `unit` *unit-number* `family inet address` *address* of a unit, other than 0, that contains the `ip-address-owner service-plane` option

1. Configure the redundant service PIC on chassis 2.

   The `redundancy-peer ipaddress` points to the address of the unit (unit 10) on ms-4/0/0 on chassis on chassis 1 that contains the `ip-address-owner service-plane` statement.

   ```
   [edit interfaces}
   set interfaces ms-4/0/0 redundancy-options redundancy-peer ipaddress 5.5.5.1
   user@host# set interfaces ms-4/0/0 redundancy-options redundancy-local data-address 5.5.5.2
   ```

```
user@host# set interfaces ms-4/0/0 redundancy-options replication-threshold 180
user@host# set interfaces ms-4/0/0 redundancy-options routing-instance HA
user@host# set interfaces ms-4/0/0 unit 10 ip-address-owner service-plane
user@host# set interfaces ms-4/0/0 unit 10 family inet address 5.5.5.2/32
user@host# set interfaces ms-4/0/0 unit 20 family inet
user@host# set interfaces ms-4/0/0 unit 20 service-domain inside
user@host# set interfaces ms-4/0/0 unit 30 family inet
user@host# set interfaces ms-4/0/0 unit 30 service-domain outside
```

2. Configure the interfaces for chassis 2 that are used as interchassis links for synchronization traffic

```
user@host# set interfaces ge-2/0/0 vlan-tagging
user@host# set interfaces ge-2/0/0 unit 0 vlan-id 100 family inet address 20.1.1.2/24
```

3. Configure remaining interfaces for chassis 2 as needed.

Results

```
user@host# show interfaces
ms-4/0/0 {
    redundancy-options {
        redundancy-peer {
            address 5.5.5.1;
        }
        redundancy-local {
            data-address 5.5.5.2;
        }
        routing-instance HA;
    }
    unit 0 {
        family inet;
    }
    unit 10 {
        ip-address-owner service-plane;
        family inet {
            address 5.5.5.2/32;
        }
    }
}
ge-2/0/0 {
    vlan-tagging;
    unit 0 {
        vlan-id 100;
        family inet {
             address 20.1.1.2/24;
        }
    }
    unit 10 {
        vlan-id 10;
        family inet {
            address 2.10.1.2/24;
        }
    }
}
```

Configure Routing Information for HA Synchronization Traffic Between MX Series Routers for Chassis 2

Step-by-Step Procedure

Detailed routing configuration is not included for this example. A routing instance is required for the HA synchronization traffic between the two chassis and is included here.

- Configure routing instances for chassis 2.

```
user@host# set routing-instances HA instance-type vrf
user@host# set routing-instances HA interface ge-2/0/0.0
user@host# set routing-instances HA interface ms-4/0/0.10
user@host# set routing-instances HA route-distinguisher 1:1
user@host# set policy-options policy-statement dummy term 1 then reject
user@host# set routing-instances HA vrf-import dummy
user@host# set routing-instances HA vrf-export dummy
user@host# set routing-instances HA routing-options static route 5.5.5.2/32 next-hop
ms-4/0/0.10
user@host# set routing-instances HA routing-options static route 5.5.5.1/32 next-hop 20.1.1.1
user@host# set routing-options static-route 100.100.100.0/24 next-hop ms-4/0/0.20
```

NOTE: The following configuration steps are *identical* to the steps shown for chassis 1.

- Configuring NAT
- Configuring the Service Set

Results

```
@user@host# show services routing-instances
HA {
    instance-type vrf;
    interface xe-2/2/0.0;
    interface ms-4/0/0.10;
    route-distinguisher 1:1;
    vrf-import dummy;
    vrf-export dummy;
    routing-options {
        static {
            route 5.5.5.2/32 next-hop ms-4/0/0.10;
            route 5.5.5.1/32 next-hop 20.1.1.1;
        }
    }
}
```

Service Redundancy Daemon

IN THIS SECTION

- Service Redundancy Daemon Overview | **891**
- Configuring the Service Redundancy Daemon | **893**
- Using Service Redundancy Daemon Scripts to View and Change the Status of a Gateway | **901**

Service Redundancy Daemon Overview

IN THIS SECTION

- Introduction to the Service Redundancy Daemon | 891
- Service Redundancy Daemon Components | 891
- Service Redundancy Daemon Constraints | 892
- Service Redundancy Daemon Operation | 893

Introduction to the Service Redundancy Daemon

- The service redundancy daemon (srd) provides configurable events that can decide when redundancy occurs across multiple gateways on MX Series routers with MS-MPCs and MS-MICs. This enables you to manage primary-role switchovers based on a monitored event. You can configure redundancy based on monitored events, including:
  - Link down events.
  - FPC and PIC reboots.
  - Routing protocol daemon (rpd) terminates and restarts.
  - Peer gateway events, including requests to acquire or release primary role, or to broadcast warnings.

Service Redundancy Daemon Components

The following configurable components control srd processing:

- **Redundancy Event**—A monitored critical event that triggers the srd to acquire or release primary role for redundancy peers, or to trigger warning-only events, and to add or delete signal routes. Monitored events include interface or link down events, rpd events, and acquire or release primary role events from peers.

- **Redundancy Policy**—A policy that defines the set of actions taken when a redundancy event occurs. Available actions include acquisition or release of primary role, and addition or deletion of signal routes.

- **Redundancy Set**—A collection of one or more service sets with a common redundancy policy or policies. A redundancy set applies to two or more system gateways. Only one of the gateways is primary and the peer or peers are standby at any time. Redundancy policies define the actions to be taken for a redundancy set when the srd detects a triggering event.

- **Redundancy Group**—A one-to-one relationship exists between a redundancy set and a redundancy group. One redundancy set can be part of only one redundancy group.
- **Signal routes**—Static routes that are added or deleted by the srd based on primary role state changes.
- **Routing Policies**—Policies that are configured to advertise routes based on the existence or non-existence of signal routes using the `if-route-exists` condition.
- **VRRP (Virtual Router Redundancy Protocol) route tracking**—TA standard Junos OS VRRP feature, but optional srd component, that tracks whether a reachable route exists in the routing table of the routing instance included in the configuration and dynamically changes the priority of the VRRP group based on the reachability of the tracked route, triggering a new primary router election. The route to be tracked is a signal route.

Service Redundancy Daemon Constraints

The following constraints apply to srd processing configurations:

- A one-to-one relationship exists between a redundancy set and a redundancy group. One redundancy set can be part of only one redundancy group.
- One redundancy policy can be part of only one redundancy set, but one redundancy set can have multiple redundancy policies. For example, redundancy set RS1 can include redundancy policies RP1 and RP2. Redundancy policies RP1 and RP2 cannot be included in redundancy sets other than RS1.
- One redundancy event can be part of only one redundancy policy, but one redundancy policy can have multiple redundancy events. For example, redundancy policy RP1 can include redundancy events RE1 and RE2. Redundancy events RE1 and RE2 cannot be included in redundancy policies other than RP1.
- One monitored interface or link can be part of only one redundancy event, but one redundancy event can have multiple monitored interfaces.
- One service set can be part of only one redundancy set, but one redundancy set may have multiple service sets.
- If gateway 1, the chassis that is configured with the lower IP address, is the primary chassis and you deactivate SRD on it, a switchover to gateway 2 occurs. If gateway 2, the chassis that is configured with the higher IP address, is the primary chassis and you deactivate SRD on it, a switchover does not occur.

- A particular redundancy-set can be active on only one gateway, but not all redundancy sets have to be active on the same gateway. For example, redundancy set A can be active on gateway 1 while redundancy set B is active on gateway 2.

Service Redundancy Daemon Operation

The srd operates as follows:

1. The srd runs on the Routing Engine. It continuously monitors configured redundancy events.
2. When a redundancy event is detected, the srd:
    a. Adds or removes signal routes specified in the redundancy policy.

    b. Switches services to the next preferred standby gateway.

    c. Updates stateful sync roles as needed.
3. Resulting route changes cause:
    a. The routing policy connected to this route to advertise routes differently.

    b. VRRP to change advertised priorities.

To summarize the switchover process:

1. A critical event occurs.
2. srd adds or removes a signal route.
3. A routing policy advertises routes differently. VRRP changes advertised priorities.
4. Services switch over to the next preferred standby gateway.
5. Stateful synchronization is updated accordingly.

NOTE: The order of routing priorities must match the order of services primary role.

Configuring the Service Redundancy Daemon

IN THIS SECTION

- Configuring Redundancy Events | 895
- Configuring Redundancy Policies | 896

- Configuring Redundancy Set and Group | 898
- Configuring Routing Policies Supporting Redundancy | 900
- Configuring Service Sets | 901

Before you configure srd processing, we recommend that you be familiar with Configuring ICCP for MC-LAG, which explains peer relationships between gateways that are enabled to exchange primary and standby roles.

You use the following configuration statements:

- `redundancy-policy` at the `[edit policy-options]` hierarchy level
- `redundancy-event` at the `[edit event-options]` hierarchy level
- `redundancy-set` at the `[edit services]` hierarchy level

The actions to be performed when configured redundancy events occur are defined in redundancy policies. Redundancy polices are associated with redundancy sets; they are analogous to rules associated with service sets. Redundancy sets are associated to redundancy groups by redundancy group IDs. Redundancy group details are defined by the underlying Inter-Chassis Communication Protocol daemon (iccpd) configuration. Service sets and redundancy sets are associated through the `redundancy-sets` statement in service sets configuration.

In the procedures that follow, redundancy events that are configured and associated with a redundancy policy. The redundancy policy is associated with a redundancy set to take appropriate action of primary-role release or primary-role acquire. If an event is associated with a policy that takes the primary-role release action, srd checks whether the redundancy peer's state is ready or warned. If the standby is in a warned state, then the primary-role release action fails. You can restore the health check and manually execute the release primary role action.

To release primary role in any case, you can either configure the policy action as `release-mastership-force` or use the `request services redundancy-set` *`redundancy-set`* `redundancy-event` *`redundancy-event`* `trigger force` command in the operational CLI. Even if your configuration specifies the `release-mastership-force` option, using the `request services redundancy-set` *`redundancy-set`* `redundancy-event` *`redundancy-event`* `trigger force` CLI command takes precedence and primary role is released. Similarly, if a redundancy event is configured with a policy with an acquire primary-role action, then srd checks the local redundancy set state. In the case of a wait state, the action fails unless you use the `request services redundancy-set` *`redundancy-set`* `redundancy-event` *`redundancy-event`* `trigger force` CLI command. We recommend that you determine why health checks fail and take action to correct the failure. After that, when the redundancy set state returns to STANDBY, then this primary-role change action succeeds.

A particular redundancy-set can be active on only one gateway, but not all redundancy sets have to be active on the same gateway. For example, redundancy set A can be active on gateway 1 while redundancy set B is active on gateway 2.

To configure srd, perform the following configuration tasks in the recommended sequence. Configurations are shown for two gateways for which primary role may change.

Configuring Redundancy Events

To configure redundancy events:

1. Configure any link-down redundancy events for the primary gateway.

```
[edit services]
user@gateway1# set event-options redundancy-event event-name monitor link-down interface-name
```

   For example:

```
[edit services]
user@gateway1# set event-options redundancy-event RELS_MSHIP_CRIT_EV monitor link-down
ms-2/3/0.0
user@gateway1# set event-options redundancy-event RELS_MSHIP_CRIT_EV monitor link-down
xe-3/0/0.0
```

2. Configure any process redundancy events for the primary gateway.

```
[edit services]
user@gateway1# set event-options redundancy-event event-name monitor process routing restart
```

   For example:

```
[edit services]
user@gateway1# set event-options redundancy-event RELS_MSHIP_CRIT_EV monitor process routing
restart
```

3. Configure any link-down redundancy events for the standby gateway.

```
[edit services]
user@gateway2# set event-options redundancy-event event-name monitor link-down interface-name
```

For example:

```
[edit services]
user@gateway2# set event-options redundancy-event WARN_EV monitor link-down ms-2/3/0.0
user@gateway2# set event-options redundancy-event WARN_EV monitor link-down xe-3/0/0.0
```

4. Configure any process redundancy events for the standby gateway.

```
[edit services]
user@gateway2# set event-options redundancy-event event-name monitor process routing restart
```

For example:

```
[edit services]
user@gateway2# set event-options redundancy-event WARN_EV monitor process routing restart
```

5. Configure any peer redundancy events for the standby gateway.

```
[edit services]
user@gateway2# set event-options redundancy-event event-name monitor peer (mastership-acquire | mastership-release)
```

For example:

```
[edit services]
user@gateway2# set event-options redundancy-event PEER_MSHIP_ACQU_EV monitor peer mastership-acquire
user@gateway2# set event-options redundancy-event PEER_MSHIP_RELS_EV monitor peer mastership-release
```

Configuring Redundancy Policies

Service redundancy policies specify actions triggered by monitored redundancy events.

To configure redundancy policies:

1. Specify a redundancy policy and redundancy event for the primary gateway. Follow the same steps for the standby gateway.

   ```
   user@gateway1# edit policy-options redundancy-policy policy-name redundancy-events [event-list] then
   ```

2. Specify an action of acquiring or releasing primary role.

   ```
   [edit policy-options redundancy-policy policy-name redundancy-events [event-list then]
   user@gateway1# set acquire-mastership
   ```

   or

   ```
   [edit policy-options redundancy-policy policy-name redundancy-events [event-list then]
   user@gateway1# set (release-mastership | release-mastership-force)
   ```

3. (Optional) Specify an action of adding a static route.

   ```
   [edit policy-options redundancy-policy policy-name redundancy-events [event-list then]
   user@gateway1# set add-static-route destination (receive | next-hop next-hop) routing-instance routing-instance
   ```

   BEST PRACTICE: We recommend using the `receive` option.

4. (Optional) Specify an action of deleting a static route.

   ```
   [edit policy-options redundancy-policy policy-name redundancy-events [event-list then]
   user@gateway1# set delete-static-route destination routing-instance routing-instance
   ```

The following example demonstrates configuring redundancy policies for two peer gateways:

```
user@gateway1# edit policy-options redundancy-policy ACQU_MSHIP_POL redundancy-events ACQU_MSHIP_MANUAL_EV then

[edit policy-options redundancy-policy ACQU_MSHIP_POL redundancy-event ACQU_MSHIP_MANUAL_EV then]
user@gateway1# set acquire-mastership add-static-route 10.45.45.0/24 receive routing-instance SGI-PRIVATE
```

```
user@gateway1# top
user@gateway1# edit policy-options redundancy-policy RELS_MSHIP_POL redundancy-events
PEER_MSHIP_ACQU_EV then

[edit policy-options redundancy-policy RELS_MSHIP_POL redundancy-events PEER_MSHIP_ACQU_EV then]
user@gateway1# set release-mastership-force delete-static-route 10.45.45.0/24 receive routing-
instance SGI-PRIVATE
```

```
user@gateway2# edit policy-options redundancy-policy RELS_MSHIP_POL redundancy-events
PEER_MSHIP_ACQU_EV then

[edit policy-options redundancy-policy ACQU_MSHIP_POL redundancy-events ACQU_MSHIP_MANUAL_EV
then]
user@gateway2# set release-mastership-force add-static-route 10.45.45.0/24 receive routing-
instance SGI-PRIVATE
user@gateway2# top
user@gateway2# edit policy-options redundancy-policy ACQU_MSHIP_POL redundancy-events
PEER_MSHIP_RELS_EV then

[edit policy-options redundancy-policy ACQU_MSHIP_POL redundancy-events PEER_MSHIP_RELS_EV then]
user@gateway2# set acquire-mastership delete-static-route 10.45.45.0/24 receive routing-instance
SGI-PRIVATE
user@gateway2# top
user@gateway2# edit policy-options redundancy-policy WARN_POL redundancy-events WARN_EV then

[edit policy-options redundancy-policy WARN_POL redundancy-events WARN_EV then]
user@gateway2# set broadcast-warning
```

Configuring Redundancy Set and Group

The redundancy group IDs that srd uses are associated with those configured for the ICCP daemon (iccpd) through the existing ICCP configuration hierarchy by using the same redundancy group ID in the configuration of the services redundancy group.

```
iccp {
    local-ip-addr 10.1.1.1;
    peer 10.2.2.2 {
        redundancy-group-id-list 1;
        liveness-detection {
            minimum-interval 1000;
        }
    }
}
```

To configure redundancy sets:

1. Specify redundancy set and group for the primary gateway.

   ```
   [edit services]
   user@gateway1# set redundancy-set redundancy-set redundancy-group redundancy-group
   ```

   For example:

   ```
   [edit services]
   user@gateway1# set redundancy-set 1 redundancy-group 1
   ```

2. Specify redundancy policies for the redundancy set.

   ```
   [edit services]
   user@gateway1# set redundancy-set redundancy-set redundancy-policy [redundancy-policy-list]
   ```

   For example:

   ```
   [edit services]
   user@gateway1# set redundancy-set 1 redundancy-policy ACQU_MSHIP_POL RELS_MSHIP_POL WARN_POL
   ```

3. Specify redundancy set and group for the peer gateway.

   ```
   [edit services]
   user@gateway2# set redundancy-set redundancy-set redundancy-group redundancy-group
   ```

   For example:

   ```
   user@gateway2# set redundancy-set 1 redundancy-group 1
   ```

4. Specify redundancy policies for the redundancy set.

   ```
   [edit services]
   user@gateway2# set redundancy-set redundancy-set redundancy-policy [redundancy-policy-list]
   ```

   For example:

   ```
   [edit services]
   user@gateway1# set redundancy-set 1 redundancy-policy [ACQU_MSHIP_POL RELS_MSHIP_POL WARN_POL]
   ```

Configuring Routing Policies Supporting Redundancy

To configure routing policies that support redundancy:

1. At the `[edit policy-options condition]` hierarchy level, use the `if-route-exists` configuration statement to set a condition based on the existence of signal routes that requires redundancy-related routing changes. Specify the routing table that is used.

   ```
   [edit policy-options condition condition-name}
   user@gateway# set if-route-exists signal-route table routing-table
   ```

   For example:

   ```
   [edit policy-options condition switchover-route-exists]
   user@gateway# set if-route-exists 10.45.45.0/24 table bgp1_table
   ```

2. At the `[edit policy-options policy-statement statement-name]` hierarchy level, specify routing changes based on the condition indicating the existence of the signal route. For BGP, routing changes typically include change to local-preference and as-path-prepend values.

   a. To change local-preference, specify `local-preference` in the `then` clause of the policy statement.

      ```
      [edit policy-options policy-statement policy-name]
      user@gateway# set term term from protocol [protocol variables] prefix-list prefix-list
      condition condition-name then local-preference preference-value accept
      ```

For example:

```
[edit policy-options policy-statement ha-export-v6-policy]
user@gateway# set term update-local-pref from protocol static bgp prefix-list ipv4-default-
route condition switchover-route-exists then local-preference 350 accept
```

b. To change `as-path-prepend` values, specify `as-path-prepend` in the `then` clause of the policy statement.

```
[edit policy-options policy-statement policy-name]
user@gateway# set term term from prefix-list prefix-list condition condition-name then as-
path-prepend [as-prepend-values] next-hop self accept
```

For example:

```
[edit policy-options policy-statement ha-export-v6-policy
user@gateway# set term update-as-prepend prefix-list ipv6-default-route condition
switchover-route-exists then as-path-prepend "64674 64674 64674 64674" next-hop self
accept
```

Configuring Service Sets

Specify stateful synchronization of services for a service set.

Specify the service set and redundancy set.

```
[edit]
user@gateway1# set services service-set service-set-name redundancy-set-id redundancy-set
```

For example:

```
[edit]
user@gateway1# set services service-set CGN4_SP-7-0-0 redundancy-set-id 1
```

Using Service Redundancy Daemon Scripts to View and Change the Status of a Gateway

You can determine the status of a gateway, disable or enable all the interfaces on the gateway, or pull services-related MIB information from the gateway by running service redundancy daemon (srd) scripts.

Before you can use these scripts, you must enable them:

- Enable the srd scripts.

  ```
  [edit]
  user@host# set system scripts op file sdg-inservice.slax
  user@host# set system scripts op file sdg-oos.slax
  user@host# set system scripts op file services-oids.slax
  user@host# set system scripts op file srd-status.slax
  user@host# set system scripts op max-datasize 512m
  ```

Use the srd scripts as the root user:

- Disable all the interfaces on the MX series router and power off the MS-MPC cards.

  a. Ensure that all local redundancy sets are in standby mode.

     ```
     root@host> show services redundancy-group
     ```

  b. Run the sdg-oos script.

     ```
     root@host> op sdg-oos
     ```

- Enable all the interfaces on the MX series router and power on the MS-MPC cards.

  ```
  root@host> op sdg-inservice
  ```

- Check the service state of a gateway.

  ```
  root@host> op srd-status
  ```

- Pull services-related MIB information from the gateway.

  ```
  root@host> op services-oids
  ```

CHAPTER 35

Configuring Inter-Chassis Stateful Synchronization for NAT and Stateful Firewall (Release 15.1 and earlier)

IN THIS CHAPTER

- Inter-Chassis High Availability for MS-MIC and MS-MPC (Release 15.1 and earlier) | **903**

Inter-Chassis High Availability for MS-MIC and MS-MPC (Release 15.1 and earlier)

IN THIS SECTION

- Inter-Chassis High Availability for Stateful Firewall and NAPT44 Overview (MS-MIC, MS-MPC) | **904**
- Configuring Inter-Chassis High Availability for Stateful Firewall and NAPT44 (MS-MPC, MS-MIC) | **905**
- Example: Inter-Chassis Stateful High Availability for NAT and Stateful Firewall (MS-MIC, MS-MPC) | **906**

NOTE: This topic applies to Junos OS release 15.1 and earlier. (For Junos OS release 16.1 and higher, see "Inter-Chassis Stateful Synchronization for Long Lived NAT and Stateful Firewall Flows (MS-MPC, MS-MIC) Overview (Release 16.1 and later)" on page 874 .)

Inter-chassis high availability supports stateful synchronization of services using a switchover to a backup services PIC on a different chassis. This topic applies to Junos OS release 15.1 and earlier. (For Junos OS release 16.1 and higher, see "Inter-Chassis Stateful Synchronization for Long Lived NAT and Stateful Firewall Flows (MS-MPC, MS-MIC) Overview (Release 16.1 and later)" on page 874 .) The feature is described in the following topics:

Inter-Chassis High Availability for Stateful Firewall and NAPT44 Overview (MS-MIC, MS-MPC)

Carrier-grade NAT (CGN) deployments can use dual-chassis implementations to provide a redundant data path and redundancy for key components in the router. Although intra-chassis high availability can be used in dual-chassis environments, it deals only with service PIC failures. If traffic is switched to a backup router due to some other failure in the router, state is lost. Inter-chassis high availability preserves state and provides redundancy using fewer service PICs than intra-chassis high availability. Only long-lived flows are synchronized between the primary and backup chassis in the high availability pair. The service PICs do not replicate state until an explicit CLI command, `request services redundancy (synchronize | no-synchronize)`, is issued to start or stop the state replication. Stateful firewall, NAPT44, and APP state information can be synchronized.

NOTE: When both the primary and backup PICs are up, replication starts immediately when the `request services redundancy command` is issued.

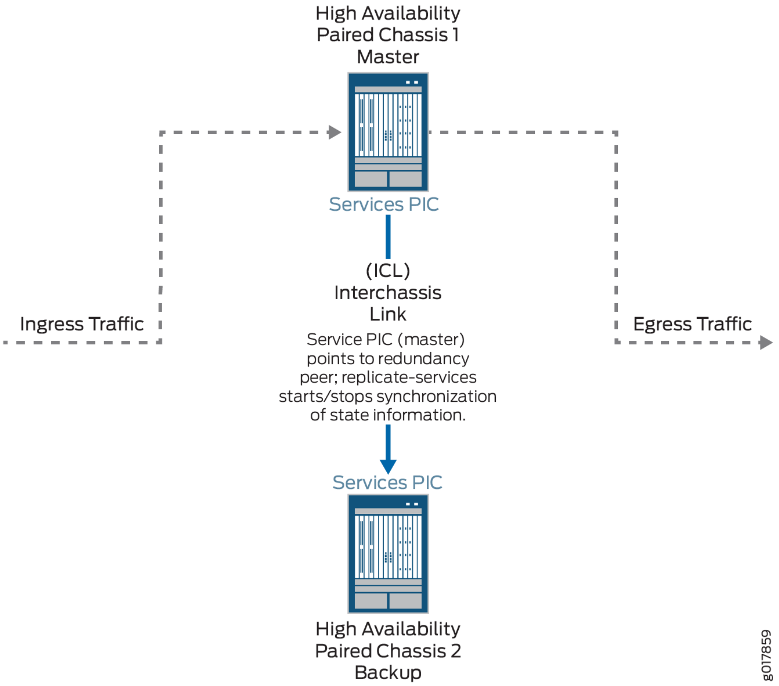
In order to use Inter-chassis high availability, you must use service sets configured for next-hop service interfaces. Inter-chassis high availability works with ms- service interfaces configured on MS-MIC or MS-MPC interface cards. A unit other than unit 0 must be configured with the `ip-address-owner service-plane` option.

The following restrictions apply:

- NAPT44 is the only translation type supported.
- Checkpointing is not supported for ALGs, PBA port block allocation (PBA), endpoint- independent mapping (EIM), or endpoint- independent filters (EIF).

Figure 44 on page 905 shows the inter-chassis high availability topology.

Figure 44: Inter-Chassis High Availability Topology



Configuring Inter-Chassis High Availability for Stateful Firewall and NAPT44 (MS-MPC, MS-MIC)

To configure inter-chassis availability for stateful firewall and NAPT44 on MS-MIC or MS-MPC service PICS, perform the following configuration steps on each chassis of the high availability pair:

1. At the `[edit interfaces` *`interface-name`* `redundancy-options]` hierarchy level, set the `ipaddress` for the `redundancy-peer`. This IPv4 address specifies one of the hosted IP addresses of the remote PIC. This address is used by the TCP channel between the HA pairs.

```
[edit interfaces interface-name redundancy-options]
user@host# set redundancy-peer ipaddress ipaddress
```

NOTE: When you enable or disable high availability of MS-MICs or MS-MPCs by configuring or removing the primary and backup adaptive services PICs by using the `redundancy-options redundancy-peer ipaddress` *`address`* statement at the `[edit interfaces` *`interface-name`*`]` hierarchy level, the configuration change is treated as a catastrophic event for each service-set that refers to the affected interface at the `[edit services service-set` *`name`* `interface-service service-interface`

interface-name] hierarchy level. A catastrophic event at the service-set level has the effect of deactivating the service set, applying the change, and then reactivating the service set.

2. Specify the name of a special routing instance, or VRF, you want applied to the HA synchronization traffic between the high availability pair.

```
[edit interfaces interface-name redundancy-options]
user@host# set routing-instance instance-name
```

3. For the service set defining an interface that is a member of the high availability pair, configure the service replication options using the `replicate-services` option.

```
[edit services service-set service-set-name  replicate-services]
user@host# set replication-threshold threshold-value
stateful-firewall
nat
```

Example: Inter-Chassis Stateful High Availability for NAT and Stateful Firewall (MS-MIC, MS-MPC)

IN THIS SECTION

- Requirements | 906
- Overview | 907
- Configuration | 907

This example shows how to configure inter-chassis high availability for stateful firewall and NAT services.

Requirements

This example uses the following hardware and software components:

- Two MX480 routers with MS-MPC line cards
- Junos OS Release 13.3 or later

Overview

Two MX 3D routers are identically configured to facilitate stateful failover for firewall and NAT services in case of a chassis failure.

Configuration

IN THIS SECTION

- CLI Quick Configuration | 907
- Configuring Interfaces for Chassis 1. | 910
- Configure Routing Information for Chassis 1 | 912
- Configuring NAT and Stateful Firewall for Chassis 1 | 913
- Configuring the Service Set | 915
- Configuring Interfaces for Chassis 2 | 916
- Configure Routing Information for Chassis 2 | 918

To configure inter-chassis high availability for this example, perform these tasks:

CLI Quick Configuration

To quickly configure this example on the routers, copy the following commands and paste them into the router terminal window after removing line breaks and substituting interface information specific to your site.

NOTE: The following configuration is for chassis 1.

```
[edit]
set interfaces ms-4/0/0 redundancy-options redundancy-peer ipaddress 5.5.5.2
set interfaces ms-4/0/0 redundancy-options routing-instance HA
set interfaces ms-4/0/0 unit 10 ip-address-owner service-plane
set interfaces ms-4/0/0 unit 10 family inet address 5.5.5.1/32
set interfaces ms-4/0/0 unit 20 family inet
set interfaces ms-4/0/0 unit 20 service-domain inside
set interfaces ms-4/0/0 unit 30 family inet
set interfaces ms-4/0/0 unit 30 service-domain outside
set interfaces ge-2/0/0 vlan-tagging
set interfaces ge-2/0/0 unit 0 vlan-id 100 family inet address 20.1.1.1/24
set routing-instances HA instance-type vrf
set routing-instances HA interface ge-2/0/0.0
set routing-instances HA interface ms-4/0/0.10
set routing-instances HA route-distinguisher 1:1
set policy-options policy-statement dummy term 1 then reject
set routing-instances HA vrf-import dummy
set routing-instances HA vrf-export dummy
set routing-instances HA routing-options static route route 5.5.5.1/32 next-hop ms-4/0/0.10
set routing-instances HA routing-options static route route 5.5.5.2/32 next-hop 20.1.1.2
set services nat pool p2 address 32.0.0.0/24
set services nat pool p2 port automatic random-allocation
set services nat pool p2 address-allocation round-robin
set services nat rule r2 match-direction input
set services nat rule r2 term t1 from source-address 129.0.0.0/8
set services nat rule r2 term t1 from source-address 128.0.0.0/8
set services nat rule r2 term t1 then translated source-pool p2
set services nat rule r2 term t1 then translated translation-type napt-44
set services nat rule r2 term t1 then translated address-pooling paired
set services nat rule r2 term t1 then syslog
set services stateful-firewall rule r2 match-direction input
set services stateful-firewall rule r2 term t1 from source-address any-unicast
set services stateful-firewall rule r2 term t1 then accept
set services stateful-firewall rule r2 term t1 then syslog
set services service-set ss2 replicate-services replication-threshold 180
set services service-set ss2 replicate-services stateful-firewall
set services service-set ss2 replicate-services nat
set services service-set ss2 stateful-firewall-rules r2
set services service-set ss2 nat-rules r2
set services service-set ss2 next-hop-service inside-service-interface ms-4/0/0.20
set services service-set ss2 next-hop-service outside-service-interface ms-4/0/0.30
set services service-set ss2 syslog host local class session-logs
set services service-set ss2 syslog host local class stateful-firewall-logs
set services service-set ss2 syslog host local class nat-logs
```

NOTE: The following configuration is for chassis 2. The NAT, stateful firewall, and service-set information must be identical for chassis 1 and 2.

```
set interfaces ms-4/0/0 redundancy-options routing-instance HA
set interfaces ms-4/0/0 unit 10 ip-address-owner service-plane
set interfaces ms-4/0/0 unit 10 family inet address 5.5.5.2/32
set interfaces ms-4/0/0 unit 20 family inet
set interfaces ms-4/0/0 unit 20 service-domain inside
set interfaces ms-4/0/0 unit 30 family inet
set interfaces ms-4/0/0 unit 30 service-domain outside
set interfaces ge-2/0/0 vlan-tagging
set interfaces ge-2/0/0 unit 0 vlan-id 100 family inet address 20.1.1.2/24
set routing-instances HA instance-type vrf
set routing-instances HA interface ge-2/0/0.0
set routing-instances HA interface ms-4/0/0.10
set routing-instances HA route-distinguisher 1:1
set routing-instances HA vrf-import dummy
set routing-instances HA vrf-export dummy
set routing-instances HA routing-options static route 5.5.5.2/32 next-hop ms-4/0/0.10
set routing-instances HA routing-options static route 5.5.5.1/32 next-hop 20.1.1.1
set services nat pool p2 address 32.0.0.0/24
set services nat pool p2 port automatic random-allocation
set services nat pool p2 address-allocation round-robin
set services nat rule r2 match-direction input
set services nat rule r2 term t1 from source-address 129.0.0.0/8
set services nat rule r2 term t1 from source-address 128.0.0.0/8
set services nat rule r2 term t1 then translated source-pool p2
set services nat rule r2 term t1 then translated translation-type napt-44
set services nat rule r2 term t1 then translated address-pooling paired
set services nat rule r2 term t1 then syslog
set services stateful-firewall rule r2 match-direction input
set services stateful-firewall rule r2 term t1 from source-address any-unicast
set services stateful-firewall rule r2 term t1 then accept
set services stateful-firewall rule r2 term t1 then syslog
set services service-set ss2 replicate-services replication-threshold 180
set services service-set ss2 replicate-services stateful-firewall
set services service-set ss2 replicate-services nat
set services service-set ss2 stateful-firewall-rules r2
set services service-set ss2 nat-rules r2
set services service-set ss2 next-hop-service inside-service-interface ms-4/0/0.20
set services service-set ss2 next-hop-service outside-service-interface ms-4/0/0.30
set services service-set ss2 syslog host local class session-logs
set services service-set ss2 syslog host local class stateful-firewall-logs
set services service-set ss2 syslog host local class nat-logs
```

Configuring Interfaces for Chassis 1.

Step-by-Step Procedure

The interfaces for each of the HA pair of routers are configured identically with the exception of the following service PIC options:

- `redundancy-options redundancy-peer ipaddress` *`address`*
- `unit` *`unit-number`* `family inet address` *`address`* of a unit, other than 0, that contains the `ip-address-owner service-plane` option

To configure interfaces:

1. Configure the redundant service PIC on chassis 1.

   ```
   [edit interfaces}
   user@host# set interfaces ms-4/0/0 redundancy-options redundancy-peer ipaddress 5.5.5.2
   user@host# set interfaces ms-4/0/0 redundancy-options routing-instance HA
   user@host# set interfaces ms-4/0/0 unit 10 ip-address-owner service-plane
   user@host# set interfaces ms-4/0/0 unit 10 family inet address 5.5.5.1/32
   user@host# set interfaces ms-4/0/0 unit 20 family inet
   user@host# set interfaces ms-4/0/0 unit 20 service-domain inside
   user@host# set interfaces ms-4/0/0 unit 30 family inet
   user@host# set interfaces ms-4/0/0 unit 30 service-domain outside
   ```

2. Configure the interfaces for chassis 1 that are used as interchassis links for synchronization traffic.

   ```
   user@host# set interfaces ge-2/0/0 vlan-tagging
   user@host# set interfaces ge-2/0/0 unit 0 vlan-id 100 family inet address 20.1.1.1/24
   ```

3. Configure remaining interfaces as needed.

Results

```
user@host# show interfaces
ge-2/0/0 {
                vlan-tagging;
        unit 0 {
            vlan-id 100;
            family inet {
                 address 20.1.1.1/24;
            }
        }
            }
ms-4/0/0 {
        redundancy-options {
            redundancy-peer {
                ipaddress 5.5.5.2;
            }
            routing-instance HA;
        }
        unit 10 {
            ip-address-owner service-plane;
            family inet {
                address 5.5.5.1/32;
            }
        }
        unit 20 {
            family inet;
            family inet6;
            service-domain inside;
        }
        unit 30 {
            family inet;
            family inet6;
            service-domain outside;
        }
            }
}
```

Configure Routing Information for Chassis 1

Step-by-Step Procedure

Detailed routing configuration is not included for this example. A routing instance is required for the HA synchronization traffic between the chassis as follows:

- Configure routing instances for Chassis 1.

```
user@host# set routing-instances HA instance-type vrf
user@host# set routing-instances HA interface ge-2/0/0.0
user@host# set routing-instances HA interface ms-4/0/0.10
user@host# set routing-instances HA route-distinguisher 1:1
user@host# set policy-options policy-statement dummy term 1 then reject
user@host# set routing-instances HA vrf-import dummy
user@host# set routing-instances HA vrf-export dummy
user@host# set routing-instances HA routing-options static route route 5.5.5.1/32 next-hop
ms-4/0/0.10
user@host# set routing-instances HA routing-options static route route 5.5.5.2/32 next-hop
20.1.1.2
```

Results

```
user@host# show routing-instances
HA {
        instance-type vrf;
        interface ge-2/0/0.0;
        interface ms-4/0/0.10;
        route-distinguisher 1:1;
        vrf-import dummy;
        vrf-export dummy;
        routing-options {
            static {
                route 5.5.5.1/32 next-hop ms-4/0/0.10;
                route 5.5.5.2/32 next-hop 20.1.1.2;
                            }
        }
    }
```

Configuring NAT and Stateful Firewall for Chassis 1

Step-by-Step Procedure

Configure NAT and stateful firewall identically on both routers. To configure NAT and stateful firewall:

1. Configure NAT as needed.

```
user@host# set services nat pool p2 address 32.0.0.0/24
user@host# set services nat pool p2 port automatic random-allocation
user@host# set services nat pool p2 address-allocation round-robin
user@host# set services nat rule r2 match-direction input
user@host# set services nat rule r2 term t1 from source-address 129.0.0.0/8
user@host# set services nat rule r2 term t1 from source-address 128.0.0.0/8
user@host# set services nat rule r2 term t1 then translated source-pool p2
user@host# set services nat rule r2 term t1 then translated translation-type napt-44
user@host# set services nat rule r2 term t1 then translated address-pooling paired
user@host# set services nat rule r2 term t1 then syslog
```

2. Configure stateful firewall as needed.

```
user@host# set services stateful-firewall rule r2 match-direction input
user@host# set services stateful-firewall rule r2 term t1 from source-address any-unicast
user@host# set services stateful-firewall rule r2 term t1 then accept
user@host# set services stateful-firewall rule r2 term t1 then syslog
```

Results

```
user@host# show services nat
nat {
        pool p2 {
            address 32.0.0.0/24;
            port {
                automatic {
                    random-allocation;
                }
            }
            address-allocation round-robin;
        }
        rule r2 {
            match-direction input;
            term t1 {
                from {
                    source-address {
                        129.0.0.0/8;
                        128.0.0.0/8;
                    }
                }
                then {
                    translated {
                        source-pool p2;
                        translation-type {
                            napt-44;
                        }
                        address-pooling paired;
                    }
                    syslog;
                }
            }
        }
    }
}
```

```
user@host show services stateful-firewell
rule r2 {
    match-direction input;
    term t1 {
        from {
            source-address {
                any-unicast;
            }
        }
        then {
            accept;
            syslog;
        }
    }
}
```

Configuring the Service Set

Step-by-Step Procedure

Configure the the service set identically on both routers. To configure the service set:

1. Configure the service set replication options.

   ```
   user@host# set services service-set ss2 replicate-services replication-threshold 180
   user@host# set services service-set ss2 replicate-services stateful-firewall
   user@host# set services service-set ss2 replicate-services nat
   ```

2. Configure references to NAT and stateful firewall rules for the service set.

   ```
   user@host# set services service-set ss2 stateful-firewall-rules r2
   user@host# set services service-set ss2 nat-rules r2
   ```

3. Configure next-hop service interface on the MS-PIC.

   ```
   user@host# set services service-set ss2 next-hop-service inside-service-interface ms-4/0/0.20
   user@host# set services service-set ss2 next-hop-service outside-service-interface ms-4/0/0.30
   ```

4. Configure desired logging options.

   ```
   user@host# set services service-set ss2 syslog host local class session-logs
   user@host# set services service-set ss2 syslog host local class stateful-firewall-logs
   user@host# set services service-set ss2 syslog host local class nat-logs
   ```

Results

```
user@host# show services service-set ss2
syslog {
            host local {
                class {
                    session-logs;
                    inactive: stateful-firewall-logs;
                    nat-logs;
                }
```

```
            }
        }
        replicate-services {
            replication-threshold 180;
            stateful-firewall;
            nat;
        }
        stateful-firewall-rules r2;
        inactive: nat-rules r2;
        next-hop-service {
            inside-service-interface ms-3/0/0.20;
            outside-service-interface ms-3/0/0.30;
        }
    }
```

Configuring Interfaces for Chassis 2

Step-by-Step Procedure

The interfaces for each of the HA pair of routers are configured identically with the exception of the following service PIC options:

- `redundancy-options redundancy-peer ipaddress` *`address`*
- `unit` *`unit-number`* `family inet address` *`address`* of a unit, other than 0, that contains the `ip-address-owner service-plane` option

1. Configure the redundant service PIC on chassis 2.

   The `redundancy-peer ipaddress` points to the address of the unit (unit 10) on ms-4/0/0 on chassis on chassis 1 that contains the `ip-address-owner service-plane` statement.

   ```
   [edit interfaces}
   set interfaces ms-4/0/0 redundancy-options redundancy-peer ipaddress 5.5.5.1
   user@host# set interfaces ms-4/0/0 redundancy-options routing-instance HA
   user@host# set interfaces ms-4/0/0 unit 10 ip-address-owner service-plane
   user@host# set interfaces ms-4/0/0 unit 10 family inet address 5.5.5.2/32
   user@host# set interfaces ms-4/0/0 unit 20 family inet
   user@host# set interfaces ms-4/0/0 unit 20 service-domain inside
   user@host# set interfaces ms-4/0/0 unit 30 family inet
   user@host# set interfaces ms-4/0/0 unit 30 service-domain outside
   ```

2. Configure the interfaces for chassis 2 that are used as interchassis links for synchronization traffic

```
user@host# set interfaces ge-2/0/0 vlan-tagging
user@host# set interfaces ge-2/0/0 unit 0 vlan-id 100 family inet address 20.1.1.2/24
```

3. Configure remaining interfaces for chassis 2 as needed.

Results

```
user@host# show interfaces
ms-4/0/0 {
        redundancy-options {
            redundancy-peer {
                ipaddress 5.5.5.1;
            }
            routing-instance HA;
        }
        unit 0 {
            family inet;
        }
        unit 10 {
            ip-address-owner service-plane;
            family inet {
                address 5.5.5.2/32;
            }
        }
ge-2/0/0 {
        vlan-tagging;
        unit 0 {
            vlan-id 100;
            family inet {
                 address 20.1.1.2/24;
            }
        }
        unit 10 {
            vlan-id 10;
            family inet {
                address 2.10.1.2/24;
            }
```

Configure Routing Information for Chassis 2

Step-by-Step Procedure

Detailed routing configuration is not included for this example. A routing instance is required for the HA synchronization traffic between the two chassis and is included here.

- Configure routing instances for chassis 2.

```
user@host# set routing-instances HA instance-type vrf
user@host# set routing-instances HA interface ge-2/0/0.0
user@host# set routing-instances HA interface ms-4/0/0.10
user@host# set routing-instances HA route-distinguisher 1:1
user@host# set policy-options policy-statement dummy term 1 then reject
user@host# set routing-instances HA vrf-import dummy
user@host# set routing-instances HA vrf-export dummy
user@host# set routing-instances HA routing-options static route 5.5.5.2/32 next-hop
ms-4/0/0.10
user@host# set routing-instances HA routing-options static route 5.5.5.1/32 next-hop 20.1.1.1
```

NOTE: The following configuration steps are *identical* to the steps shown for chassis 1.

- Configuring NAT and Stateful Firewall
- Configuring the Service Set

Results

```
user@host# show services routing-instances
HA {
        instance-type vrf;
        interface xe-2/2/0.0;
        interface ms-4/0/0.10;
        route-distinguisher 1:1;
        vrf-import dummy;
        vrf-export dummy;
        routing-options {
            static {
                route 5.5.5.2/32 next-hop ms-4/0/0.10;
                route 5.5.5.1/32 next-hop 20.1.1.1;
            }
        }
```

SEE ALSO

Configuring Inter-Chassis High Availability for Stateful Firewall and NAPT44 (MS-MPC, MS-MIC) | 905

10

PART

Multilinks

Link Services Interface Redundancy | 921

Link Bundling | 939

CHAPTER 36

Link Services Interface Redundancy

IN THIS CHAPTER

- Link Services Interface Redundancy | 921

Link Services Interface Redundancy

IN THIS SECTION

- Layer 2 Service Package Capabilities and Interfaces | 921
- Configuring LSQ Interface Redundancy Across Multiple Routers Using SONET APS | 923
- Configuring LSQ Interface Redundancy in a Single Router Using SONET APS | 926
- Configuring LSQ Interface Redundancy in a Single Router Using Virtual Interfaces | 927

Layer 2 Service Package Capabilities and Interfaces

As described in *Enabling Service Packages*, you can configure the AS or Multiservices PIC and the internal ASM in the M7i platform to use either the Layer 2 or the Layer 3 service package.

When you enable the Layer 2 service package, the AS or Multiservices PIC supports *link services*. On the AS or Multiservices PIC and the ASM, link services include the following:

- Junos CoS components—"Configuring CoS Scheduling Queues on Logical LSQ Interfaces" on page 849 describes how the Junos CoS components work on link services IQ (`lsq`) interfaces. For detailed information about Junos CoS components, see the Class of Service User Guide (Routers and EX9200 Switches).

- Data compression using the compressed Real-Time Transport Protocol (CRTP) for use in voice over IP (VoIP) transmission.

NOTE: On LSQ interfaces, all multilink traffic for a single bundle is sent to a single processor. If CRTP is enabled on the bundle, it adds overhead to the CPU. Because T3 network interfaces support only one link per bundle, make sure you configure a fragmentation map for compressed traffic on these interfaces and specify the `no-fragmentation` option. For more information, see "Configuring Delay-Sensitive Packet Interleaving" on page 1047 and "Configuring CoS Fragmentation by Forwarding Class on LSQ Interfaces" on page 853 .

- Link fragment interleaving (LFI) on Frame Relay links using FRF.12 end-to-end fragmentation—The standard for FRF.12 is defined in the specification FRF.12, *Frame Relay Fragmentation Implementation Agreement*.

- LFI on Multilink Point-to-Point Protocol (MLPPP) links.

- Multilink Frame Relay (MLFR) end-to-end (FRF.15)—The standard for FRF.15 is defined in the specification FRF.15, *End-to-End Multilink Frame Relay Implementation Agreement*.

- Multilink Frame Relay (MLFR) UNI NNI (FRF.16)—The standard for FRF.16 is defined in the specification FRF.16.1, *Multilink Frame Relay UNI/NNI Implementation Agreement*.

- MLPPP—The standard for MLPPP is defined in the specification RFC 1990, *The PPP Multilink Protocol (MP)*.

- Multiclass extension to MLPPP—The standard is defined in the specification RFC 2686, *The Multi-Class Extension to Multi-Link PPP*.

For the LSQ interface on the AS or Multiservices PIC, the configuration syntax is almost the same as for Multilink and Link Services PICs. The primary difference is the use of the interface-type descriptor `lsq` instead of `ml` or `ls`. When you enable the Layer 2 service package on the AS or Multiservices PIC, the following interfaces are automatically created:

```
gr-fpc/pic/port
ip-fpc/pic/port
lsq-fpc/pic/port
lsq-fpc/pic/port:0
...
lsq-fpc/pic/port:N
mt-fpc/pic/port
pd-fpc/pic/port
pe-fpc/pic/port
sp-fpc/pic/port
vt-fpc/pic/port
```

Interface types `gr`, `ip`, `mt`, `pd`, `pe`, and `vt` are standard tunnel interfaces that are available on the AS or Multiservices PIC whether you enable the Layer 2 or the Layer 3 service package. These tunnel interfaces function the same way for both service packages, except that the Layer 2 service package does not support some tunnel functions, as shown in Table 5 on page 24. For more information about tunnel interfaces, see Tunnel and Encryption Services Interfaces User Guide for Routing Devices.

> **NOTE**: Interface type `sp` is created because it is needed by the Junos OS. For the Layer 2 service package, the `sp` interface is not configurable, but you should not disable it.

Interface type `lsq-`*fpc*/*pic*/*port* is the physical link services IQ interface (`lsq`). Interface types `lsq-`*fpc*/*pic*/*port*`:0` through `lsq-`*fpc*/*pic*/*port*`:`*N* represent FRF.16 bundles. These interface types are created when you include the `mlfr-uni-nni-bundles` statement at the `[edit chassis fpc` *slot-number* `pic` *pic-number*`]` hierarchy level. For more information, see "Configuring CoS Scheduling Queues on Logical LSQ Interfaces" on page 849 .

> **NOTE**: On DS0, E1, or T1 interfaces in LSQ bundles, you can configure the `bandwidth` statement, but the router does not use the bandwidth value if the interfaces are included in an MLPPP or MLFR bundle. The bandwidth is calculated internally according to the time slots, framing, and byte-encoding of the interface. For more information about these properties, see the Junos OS Network Interfaces Library for Routing Devices.

Configuring LSQ Interface Redundancy Across Multiple Routers Using SONET APS

IN THIS SECTION

- Configuring the Association between LSQ and SONET Interfaces | 924
- Configuring SONET APS Interoperability with Cisco Systems FRF.16 | 925
- Restrictions on APS Redundancy for LSQ Interfaces | 925

Link services IQ (`lsq-`) interfaces that are paired with SONET PICs can use the Automatic Protection Switching (APS) configuration already available on SONET networks to provide failure recovery. SONET APS provides stateless failure recovery, if it is configured on SONET interfaces in separate chassis and each SONET PIC is paired with an AS or Multiservices PIC in the same chassis. If one of the following conditions for APS failure is met, the associated SONET PIC triggers recovery to the backup circuit and its associated AS or Multiservices PIC. The failure conditions are:

- Failure of Link Services IQ PIC

- Failure of FPC that hosts the Link Services IQ PIC
- Failure of Packet Forwarding Engine
- Failure of chassis

The guidelines for configuring SONET APS are described in the Junos OS Network Interfaces Library for Routing Devices.

The following sections describe how to configure failover properties:

Configuring the Association between LSQ and SONET Interfaces

To configure the association between AS or Multiservices PICs hosting link services IQ interfaces and the SONET interfaces, include the `lsq-failure-options` statement at the `[edit interfaces]` hierarchy level:

```
lsq-fpc/pic/port {
    lsq-failure-options {
        no-termination-request;
        [ trigger-link-failure interface-name ];
    }
}
```

For example, consider the following network scenario:

- Primary router includes interfaces `oc3-0/2/0` and `lsq-1/1/0`.
- Backup router includes interfaces `oc3-2/2/0` and `lsq-3/2/0`.

Configure SONET APS, with `oc3-0/2/0` as the working circuit and `oc3-2/2/0` as the protect circuit. Include the `trigger-link-failure` statement to extend failure to the LSQ PICs:

```
interfaces lsq-1/1/0 {
    lsq-failure-options {
        trigger-link-failure oc3-0/2/0;
    }
}
```

NOTE: You must configure the `lsq-failure-options` statement on the primary router only. The configuration is not supported on the backup router.

To inhibit the router from sending PPP termination-request messages to the remote host if the Link Services IQ PIC fails, include the `no-termination-request` statement at the `[edit interfaces lsq-*fpc/pic/port* lsq-failure-options]` hierarchy level:

```
[edit interfaces lsq-fpc/pic/port lsq-failure-options]
no-termination-request;
```

This functionality is supported on link PICs as well. To inhibit the router from sending PPP termination-request messages to the remote host if a link PIC fails, include the `no-termination-request` statement at the `[edit interfaces *interface-name* ppp-options]` hierarchy level.

```
[edit interfaces interface-name ppp-options]
no-termination-request;
```

The `no-termination-request` statement is supported only with MLPPP and SONET APS configurations and works with PPP, PPP over Frame Relay, and MLPPP interfaces only, on the following PICs:

- Channelized OC3 IQ PICs
- Channelized OC12 IQ PICs
- Channelized STM1 IQ PICs
- Channelized STM4 IQ PICs

Configuring SONET APS Interoperability with Cisco Systems FRF.16

Juniper Networks routers configured with APS might not interoperate correctly with Cisco FRF.16. To enable interoperation, include the `cisco-interoperability` statement at the `[edit interfaces lsq-*fpc/pic/port* mlfr-uni-nni-bundle-options]` hierarchy level:

```
[edit interfaces lsq-fpc/pic/port mlfr-uni-nni-bundle-options]
cisco-interoperability send-lip-remove-link-for-link-reject;
```

The `send-lip-remove-link-for-link-reject` option prompts the router to send a Link Integrity Protocol remove link when it receives an add-link rejection message.

Restrictions on APS Redundancy for LSQ Interfaces

The following restrictions apply to LSQ failure recovery:

- It applies only to Link Services IQ PICs installed in M Series routers, except for M320 routers.

- You must configure the `failure-options` statement on physical LSQ interfaces, not on MLFR channelized units.
- The Link Services IQ PICs must be associated with SONET link PICs. The paired PICs can be installed on different routers or in the same router; in other words, both interchassis and intrachassis recovery are supported
- Failure recovery is stateless; as a result, route flapping and loss of link state is expected in interchassis recovery, requiring PPP renegotiation. In intrachassis recovery, no impact on traffic is anticipated with Routing Engine failover, but PIC failover results in PPP renegotiation.
- The switchover is not revertive: when the original hardware is restored to service, traffic does not automatically revert back to it.
- Normal APS switchover and PIC-triggered APS switchover can be distinguished only by checking the system log messages.

> **NOTE**: When an AS PIC experiences persistent back pressure as a result of high traffic volume for 3 seconds, the condition triggers an automatic core dump and reboot of the PIC to help clear the blockage. A system log message at level LOG_ERR is generated. This mechanism applies to both Layer 2 and Layer 3 service packages.

SEE ALSO

Configuring Link Services and CoS on Services PICs | **856**

Link Services Configuration for Junos Interfaces | **847**

Configuring LSQ Interface Redundancy in a Single Router Using SONET APS

Stateless switchover from one Link Services IQ PIC to another within the same router can be configured by using the SONET APS mechanism described in "Configuring LSQ Interface Redundancy Across Multiple Routers Using SONET APS" on page 921 . Each Link Services IQ PIC must be associated with a specified SONET link PIC within the same router.

> **NOTE**: For complete intrachassis recovery, including recovery from Routing Engine failover, graceful Routing Engine switchover (GRES) must be enabled on the router. For more information, see the Junos OS Administration Library for Routing Devices.

SEE ALSO

Configuring Link Services and CoS on Services PICs | 856

Link Services Configuration for Junos Interfaces | 847

Configuring LSQ Interface Redundancy in a Single Router Using Virtual Interfaces

IN THIS SECTION

- Configuring Redundant Paired LSQ Interfaces | 927
- Restrictions on Redundant LSQ Interfaces | 929
- Configuring Link State Replication for Redundant Link PICs | 930
- Examples: Configuring Redundant LSQ Interfaces for Failure Recovery | 932

You can configure failure recovery on M Series, MX Series, and T Series routers that have multiple AS or Multiservices PICs and DPCs with `lsq-` interfaces by specifying a virtual LSQ redundancy (`rlsq`) interface in which the primary Link Services IQ PIC is active and a secondary PIC is on standby. If the primary PIC fails, the secondary PIC becomes active, and all LSQ processing is transferred to it. To determine which PIC is currently active, issue the `show interfaces redundancy` command.

NOTE: This configuration does not require the use of SONET APS for failover. Network interfaces that do not support SONET can be used, such as T1 or E1 interfaces.

The following sections provide more information:

Configuring Redundant Paired LSQ Interfaces

The physical interface type `rlsq` specifies the pairings between primary and secondary `lsq` interfaces to enable redundancy. To configure a backup `lsq` interface, include the `redundancy-options` statement at the `[edit interfaces rlsq`*`number`*`]` hierarchy level:

```
[edit interfaces rlsqnumber]
redundancy-options {
    (hot-standby | warm-standby);
    primary lsq-fpc/pic/port;
    secondary lsq-fpc/pic/port;
}
```

For the rlsq interface, *number* can be from 0 through 1023. If the primary lsq interface fails, traffic processing switches to the secondary interface. The secondary interface remains active even after the primary interface recovers. If the secondary interface fails and the primary interface is active, processing switches to the primary interface.

The hot-standby option is used with one-to-one redundancy configurations, in which one working PIC is supported by one backup PIC. It is supported with MLPPP, CRTP, FRF.15, and FRF.16 configurations for the LSQ interface to achieve an uninterrupted LSQ service. It sets the requirement for the failure detection and recovery time to be less than 5 seconds. The behavior is revertive, but you can manually switch between the primary and secondary PICs by issuing the request interfaces (revert | switchover) rlsq*number* operational mode command. It also provides a switch over time of 5 seconds and less for FRF.15 and a maximum of 10 seconds for FRF.16.

The warm-standby option is used with redundancy configurations in which one backup PIC supports multiple working PICs. Recovery times are not guaranteed, because the configuration must be completely restored on the backup PIC after a failure is detected.

Certain combinations of hot-standby and warm-standby configuration are not permitted and result in a configuration error. The following examples are permitted:

- Interface rlsq0 configured with primary lsq-0/0/0 and warm-standby, in combination with interface rlsq0:0 configured with primary lsq-0/0/0:0
- Interface rlsq0:0 configured with primary lsq-0/0/0:0, in combination with interface rlsq0:1 configured with primary lsq-0/0/0:1

The following example combinations are not permitted:

- Interface rlsq0 configured with primary lsq-0/0/0 and hot-standby, in combination with interface rlsq0:0 configured with primary lsq-0/0/0:0
- Interface rlsq0:0 configured with primary lsq-0/0/0:0, in combination with interface rlsq1:0 configured with primary lsq-0/0/0:0
- Interface rlsq0:0 configured with primary lsq-0/0/0:1, in combination with interface rlsq1:1 configured with primary lsq-0/0/0:1
- Interface rlsq0 configured with primary lsq-0/0/0, in combination with interface rlsq1 configured with primary lsq-0/0/0

In addition, the same physical interface cannot be reused as the primary interface for more than one rlsq interface, nor can any of the associated logical interfaces. For example, primary interface lsq-0/0/0 cannot be reused in another rlsq interface as lsq-0/0/0:0.

Restrictions on Redundant LSQ Interfaces

Link Services IQ PIC failure occurs under the following conditions:

- The primary PIC fails to boot. In this case, the `rlsq` interface does not come up and manual intervention is necessary to reboot or replace the PIC, or to rename the primary PIC to the secondary one in the `rlsq` configuration.
- If the following conditions are not met when configuring an `rlsq` interface:
  - The unit number allocated to the `rlsq` interface is less than the number of Multilink Frame Relay user-to-network interface network-to-network interface (UNI-NNI) (FRF.16) bundles allocated on the Link Services PIC.
  - Data-link connection identifier (DLCI) is configured for the `rlsq` interface.

  If these conditions are not met, the `rlsq` interface does not boot. When you issue the `show interfaces redundancy` command, the state of the `rlsq` interface is indicated as `Waiting for primary MS PIC`.
- The primary PIC becomes active and then fails. The secondary PIC automatically takes over processing.
- A failover to the secondary PIC takes place. The secondary PIC then fails. If the primary PIC has been restored to active state, processing switches to it.
- The FPC that contains the Link Services IQ PIC fails.

The following constraints apply to redundant LSQ configurations:

- We recommend that primary and secondary PICs be configured in two different FPCs (in chassis other than M10i routers).
- You cannot configure a Link Services IQ PIC with explicit bundle configurations and as a constituent of an `rlsq` interface.
- Redundant LSQ configurations provide full GRES support. (You must configure GRES at the `[edit chassis]` hierarchy level; see the Junos OS Administration Library for Routing Devices.
- If you configure the `redundancy-options` statement with the `hot-standby` option, the configuration must include one `primary` interface value and one `secondary` interface value.
- Since the same interface name is used for `hot-standby` and `warm-standby`, if you modify the configuration to change this attribute, it is recommended that you first deactivate the interface, commit the new configuration, and then reactivate the interface.
- You cannot make changes to an active `redundancy-options` configuration. You must deactivate the `rlsq`*`number`* interface configuration, change it, and reactivate it.

- The rlsq*number* configuration becomes active only if the primary interface is active. When the configuration is first activated, the primary interface must be active; if not, the rlsq interface waits until the primary interface comes up.

- You cannot modify the configuration of lsq interfaces after they have been included in an active rlsq interface.

- All the operational mode commands that apply to rsp interfaces also apply to rlsq interfaces. You can issue show commands for the rlsq interface or the primary and secondary lsq interfaces. However, statistics on the link interfaces are not carried over following a Routing Engine switchover.

- The rlsq interfaces also support the lsq-failure-options configuration, discussed in "Configuring LSQ Interface Redundancy Across Multiple Routers Using SONET APS" on page 921 . If the primary and secondary Link Services IQ PICs fail and the lsq-failure-options statement is configured, the configuration triggers a SONET APS switchover.

- Redundant LSQ configurations that require MLPPP Multilink Frame Relay (FRF.15 and FRF.16) are supported only with the warm-standby option.

- Redundant LSQ support is extended to ATM network interfaces.

- Channelized interfaces are used with FRF-16 bundles, for example rlsq0:0. The rlsq number and its constituents, the primary and secondary interfaces, must match for the configuration to be valid: either all must be channelized, or none. For an example of an FRF.16 configuration, see "No Link Title" on page 937 .

- When you configure a channelized rlsq interface, you must use a channel index number from 0 through 254.

> **NOTE**: Adaptive Services and Multiservices PICs in layer-2 mode (running Layer 2 services) are not rebooted when a MAC flow-control situation is detected.

Configuring Link State Replication for Redundant Link PICs

Link state replication, also called *interface preservation*, is an addition to the SONET Automatic Protection Switching (APS) functionality that helps promote redundancy of the link PICs used in LSQ configurations.

Link state replication provides the ability to add two sets of links, one from the active (working) SONET PIC and the other from the backup (protect) SONET PIC to the same bundle. If the active SONET PIC fails, links from the standby PIC are used without causing a link renegotiation. All the negotiated state is replicated from the active links to the standby links to prevent link renegotiation. For more information about SONET APS configurations, see the Junos OS Network Interfaces Library for Routing Devices.

To configure link state replication, include the `preserve-interface` statement at the `[edit interfaces` *`interface-name`* `sonet-options aps]` hierarchy level on both network interfaces:

```
edit interfaces interface-name sonet-options aps]
preserve-interface;
```

The following constraints apply to link PIC redundancy:

- APS functionality must be available on the SONET PICs and the interface configurations must be identical on both ends of the link. Any configuration mismatch causes the commit operation to fail.

- This feature is supported only with LSQ and SONET APS-enabled link PICs, including Channelized OC3, Channelized OC12, and Channelized STM1 intelligent queuing (IQ) PICs.

- Link state replication supports MLPPP and PPP over Frame Relay (`frame-relay-ppp`) encapsulation, and fully supports GRES.

- Enabling the interface or protocol traceoptions with a large number of MLPPP links can trigger Link Control Protocol (LCP) renegotiation during the link switchover time.

  NOTE: This renegotiation is more likely to take place for configurations with back-to-back Juniper Networks routers than in networks in which a Juniper Networks router is connected to an add/drop multiplexer (ADM).

- In general, networks that connect a Juniper Networks router to an ADM allow faster MLPPP link switchover than those with back-to-back Juniper Networks routers. The MLPPP link switchover time difference may be significant, especially for networks with a large number of MLPPP links.

- An aggressive LCP keepalive timeout configuration can lead to LCP renegotiation during the MLPPP link switchover. By default, the LCP keepalive timer interval is 10 seconds and the consecutive link down count is 3. The MLPPP links start LCP negotiation only after a timeout of 30 seconds. Lowering these configuration values may trigger one or more of the MLPPP links to renegotiate during the switchover time.

  NOTE: LCP renegotiation is more likely to take place for configurations with back-to-back Juniper Networks routers than in networks in which a Juniper Networks router is connected to an ADM.

As an example, the following configuration shows the link state replication configuration between the ports coc3-1/0/0 and coc3-2/0/0.

```
interfaces {
    coc3-1/0/0 {
        sonet-options {
            aps {
                preserve-interface;
                working-circuit aps-group-1;
            }
        }
    }
    coc3-2/0/0 {
        sonet-options {
            aps {
                preserve-interface;
                protect-circuit aps-group-1;
            }
        }
    }
}
```

Examples: Configuring Redundant LSQ Interfaces for Failure Recovery

Configuring LSQ Interface Redundancy for MLPPP

The following configuration shows that lsq-1/1/0 and lsq-1/3/0 work as a pair and the redundancy type is hot-standby, which sets the requirement for the failure detection and recovery time to be less than 5 seconds:

```
interfaces rlsq0 {
    redundancy-options {
        primary lsq-1/1/0;
        secondary lsq-1/3/0;
        hot-standby; #either hot-standby or warm-standby is supported
    }
}
```

The following example shows a related MLPPP configuration:

NOTE: MLPPP protocol configuration is required for this configuration.

```
interfaces {
    t1-/1/2/0 {
        unit 0 {
            family mlppp {
                bundle rlsq0.0;
            }
        }
    }
    rlsq0 {
        unit 0 {
            family inet {
                address 10.30.1.2/24;
            }
        }
    }
}
```

The following example shows a related CoS configuration:

```
class-of-service {
    interfaces {
        rlsq0 {
            unit * {
                fragmentation-maps fr-map1;
            }
        }
    }
}
```

The following example shows a complete link state replication configuration for MLPPP. This example uses two bundles, each with four T1 links. The first four T1 links (`t1-*:1` through `t1-*:4`) form the first bundle and the last four T1 links (`t1-*:5` through `t1-*:8`) form the second bundle. To minimize the duplication in the configuration, this example uses the `[edit groups]` statement; for more information, see the Junos OS Administration Library for Routing Devices. This type of configuration is not required; it simplifies the task and minimizes duplication.

```
groups {
    ml-partition-group {
        interfaces {
            <coc3-*> {
                partition 1 oc-slice 1 interface-type coc1;
            }
            <coc1-*> {
                partition 1-8 interface-type t1;
            }
        }
    }
    ml-bundle-group-1 {
        interfaces {
            <t1-*:"[1-4]"> {
                encapsulation ppp;
                unit 0 {
                    family mlppp {
                        bundle lsq-0/1/0.0;
                    }
                }
            }
        }
    }
    ml-bundle-group-2 {
        interfaces {
            <t1-*:"[5-8]"> {
                encapsulation ppp;
                unit 0 {
                    family mlppp {
                        bundle lsq-0/1/0.1;
                    }
                }
            }
        }
    }
}
interfaces {
    lsq-0/1/0 {
        unit 0 {
            encapsulation multilink-ppp;
            family inet {
                address 10.1.1.1/32 {
                    destination 10.1.1.2;
                }
            }
        }
        unit 1 {
            encapsulation multilink-ppp;
            family inet {
                address 10.1.2.1/32 {
                    destination 10.1.2.2;
                }
            }
        }
    }
    coc3-1/0/0 {
        apply-groups ml-partition-group;
        sonet-options {
            aps {
                preserve-interface;
                working-circuit aps-group-1;
            }
        }
    }
    coc1-1/0/0:1 {
        apply-groups ml-partition-group;
    }
    t1-1/0/0:1:1 {
        apply-groups ml-bundle-group-1;
    }
    t1-1/0/0:1:2 {
        apply-groups ml-bundle-group-1;
    }
    t1-1/0/0:1:3 {
        apply-groups ml-bundle-group-1;
    }
    t1-1/0/0:1:4 {
        apply-groups ml-bundle-group-1;
    }
    t1-1/0/0:1:5 {
        apply-groups ml-bundle-group-2;
    }
    t1-1/0/0:1:6 {
        apply-groups ml-bundle-group-2;
    }
    t1-1/0/0:1:7 {
        apply-groups ml-bundle-group-2;
    }
    t1-1/0/0:1:8 {
        apply-groups ml-bundle-group-2;
    }
    coc3-2/0/0 {
        apply-groups ml-partition-group;
        sonet-options {
            aps {
                preserve-interface;
                protect-circuit aps-group-1;
            }
        }
    }
    coc1-2/0/0:1 {
        apply-groups ml-partition-group;
    }
    t1-2/0/0:1:1 {
        apply-groups ml-bundle-group-1;
    }
    t1-2/0/0:1:2 {
        apply-groups ml-bundle-group-1;
    }
    t1-2/0/0:1:3 {
        apply-groups ml-bundle-group-1;
    }
    t1-2/0/0:1:4 {
        apply-groups ml-bundle-group-1;
    }
    t1-2/0/0:1:5 {
        apply-groups ml-bundle-group-2;
    }
    t1-2/0/0:1:6 {
        apply-groups ml-bundle-group-2;
    }
    t1-2/0/0:1:7 {
        apply-groups ml-bundle-group-2;
    }
    t1-2/0/0:1:8 {
        apply-groups ml-bundle-group-2;
    }
}
```

Configuring LSQ Interface Redundancy for an FRF.15 Bundle

The following example shows a configuration for an FRF.15 bundle:

```
interfaces rlsq0 {
    redundancy-options {
        primary lsq-1/2/0;
        secondary lsq-1/3/0;
        warm-standby; #either hot-standby or warm-standby is supported
    }
    unit 0 {
        encapsulation multilink-frame-relay-end-to-end;
        family inet {
            address 10.30.1.1/24;
        }
    }
}
```

Configuring LSQ Interface Redundancy for an FRF.16 Bundle

The following example shows a configuration for an FRF.16 bundle:

```
interfaces rlsq0:0 {
    dce;
    encapsulation multilink-frame-relay-uni-nni;
    redundancy-options {
        primary lsq-1/2/0:0;
        secondary lsq-1/3/0:0;
        warm-standby; #either hot-standby or warm-standby is supported
    }
    unit 0 {
        dlci 1000;
        family inet {
            address 10.50.1.1/24;
        }
    }
}
```

SEE ALSO

Configuring Link Services and CoS on Services PICs | 856

Link Services Configuration for Junos Interfaces | 847

CHAPTER 37

Link Bundling

IN THIS CHAPTER

- Inline Multlink Services | **939**

Inline Multlink Services

IN THIS SECTION

- Inline MLPPP for WAN Interfaces Overview | **939**
- Reserving Bundle Bandwidth for Link-Layer Overhead on LSQ Interfaces | **942**
- Enabling Inline LSQ Services | **943**
- Configuring LSQ Interfaces as NxT1 or NxE1 Bundles Using MLPPP | **945**
- Configuring LSQ Interfaces as NxT1 or NxE1 Bundles Using FRF.16 | **952**
- Configuring LSQ Interfaces as NxT1 or NxE1 Bundles Using FRF.15 | **959**
- Configuring LSQ Interfaces for Single Fractional T1 or E1 Interfaces Using MLPPP and LFI | **960**
- Configuring LSQ Interfaces for Single Fractional T1 or E1 Interfaces Using FRF.12 | **966**
- Configuring LSQ Interfaces for T3 Links Configured for Compressed RTP over MLPPP | **976**
- Configuring LSQ Interfaces as T3 or OC3 Bundles Using FRF.12 | **977**
- Configuring LSQ Interfaces for ATM2 IQ Interfaces Using MLPPP | **980**

Inline MLPPP for WAN Interfaces Overview

Inline Multilink PPP (*MLPPP*), Multilink Frame Relay (*FRF.16*), and Multilink Frame Relay End-to-End (*FRF.15*) for time-division multiplexing (*TDM*) WAN interfaces provide bundling services through the Packet Forwarding Engine without requiring a PIC or Dense Port Concentrator (*DPC*).

Traditionally, bundling services are used to bundle multiple low-speed links to create a higher bandwidth pipe. This combined bandwidth is available to traffic from all links and supports link fragmentation and interleaving (*LFI*) on the bundle, reducing high priority packet transmission delay.

This support includes multiple links on the same bundle as well as multiclass extension for MLPPP. Through this service you can enable bundling services without additional DPC slots to support Service DPC and free up the slots for other MICs.

NOTE: MLPPP is not supported on MX Series Virtual Chassis.

Starting in Junos OS Release 15.1, you can configure inline MLPPP interfaces on MX80, MX104, MX240, MX480, and MX960 routers with Channelized E1/T1 Circuit Emulation MICs. A maximum of up to eight inline MLPPP interface bundles are supported on Channelized E1/T1 Circuit Emulation MICs, similar to the support for inline MLPPP bundles on other MICs with which they are compatible.

Configuring inline MLPPP for WAN interfaces benefits the following services:

- CE-PE link for Layer 3 VPN and DIA service with public switched telephone networks (PSTN)-based access networks.

- PE-P link when PSTN is used for MPLS networks.

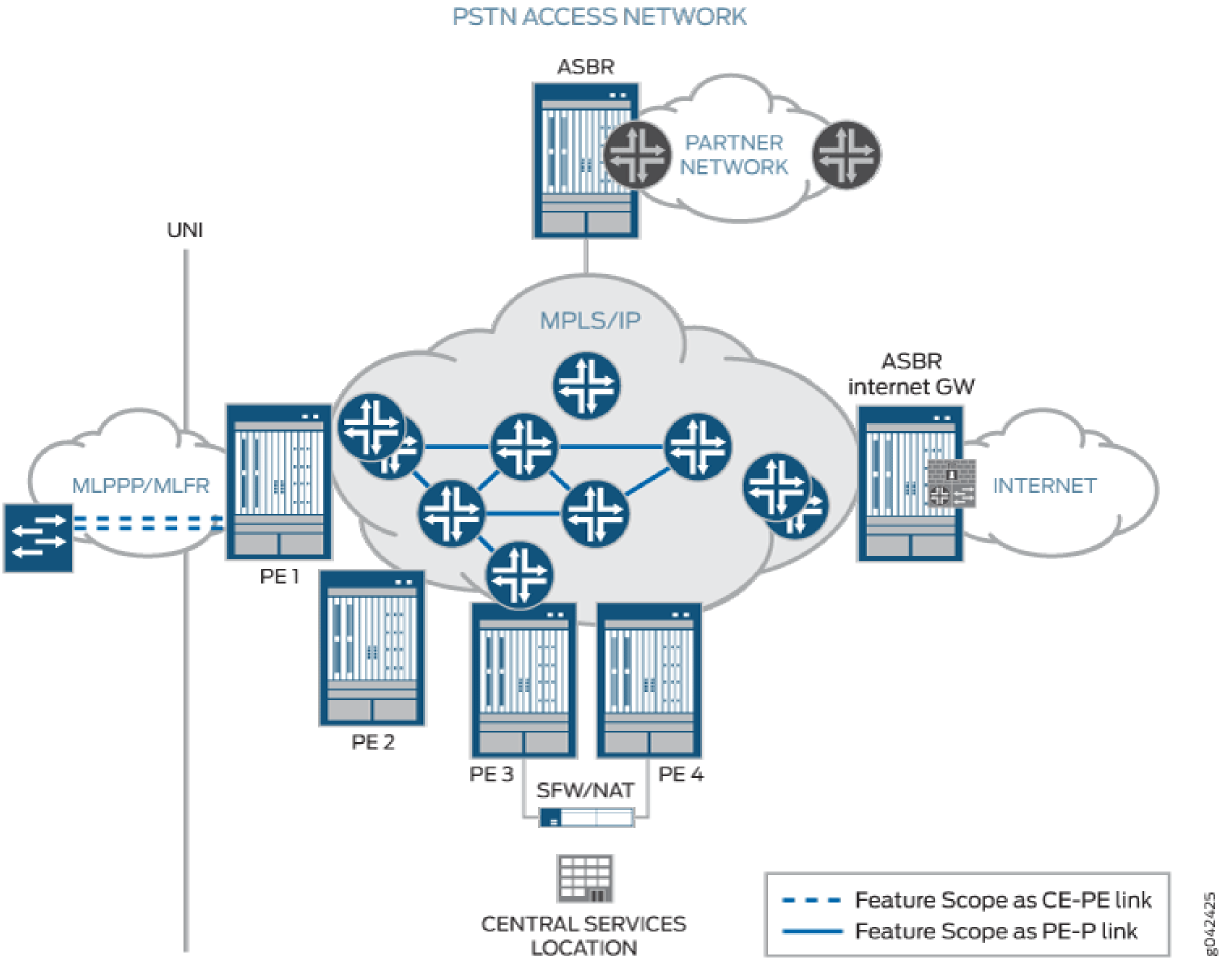
This feature is used by the following service providers:

- Service providers that use PSTN to offer Layer 3 VPN and DIA service with PSTN-based access networks to medium or large business customers.

- Service providers with SONET-based core networks.

The following figure illustrates the scope of this feature:

Figure 45: Inline MLPPP for WAN Interfaces



For connecting many smaller sites in VPNs, bundling the TDM circuits together with MLPPP/MLFR technology is the *only* way to offer higher bandwidth and link redundancy.

MLPPP enables you to bundle multiple PPP links into a single multilink bundle, and MLFR enables you to bundle multiple Frame Relay data-link connection identifiers (DLCIs) into a single multilink bundle. Multilink bundles provide additional bandwidth, load balancing, and redundancy by aggregating low-speed links, such as T1, E1, and serial links.

MLPPP is a protocol for aggregating multiple constituent links into one larger PPP bundle. MLFR allows you to aggregate multiple Frame Relay links by inverse multiplexing. MLPPP and MLFR provide service options between low-speed T1 and E1 services. In addition to providing additional bandwidth, bundling multiple links can add a level of fault tolerance to your dedicated access service. Because you can implement bundling across multiple interfaces, you can protect users against loss of access when a single interface fails.

To configure inline MLPPP for WAN interfaces, see:

- *Example: Configuring Inline MLPPP and Multilink Frame Relay End-to-End (FRF.15) for WAN Interfaces*
- *Example: Configuring Inline Multilink Frame Relay (FRF.16) for WAN Interfaces*

SEE ALSO

Enabling MLPPP Link Fragmentation and Interleaving

Example: Configuring Multilink Frame Relay FRF.15

Example: Configuring Multilink Frame Relay FRF.16

https://www.juniper.net/documentation/en_US/junos/information-products/pathway-pages/services-interfaces/link-multilink-properties.html

Reserving Bundle Bandwidth for Link-Layer Overhead on LSQ Interfaces

Link-layer overhead can cause packet drops on constituent links because of bit stuffing on serial links. Bit stuffing is used to prevent data from being interpreted as control information.

By default, 4 percent of the total bundle bandwidth is set aside for link-layer overhead. In most network environments, the average link-layer overhead is 1.6 percent. Therefore, we recommend 4 percent as a safeguard. For more information, see RFC 4814, *Hash and Stuffing: Overlooked Factors in Network Device Benchmarking*.

For link services IQ (`lsq-`) interfaces, you can configure the percentage of bundle bandwidth to be set aside for link-layer overhead. To do this, include the `link-layer-overhead` statement:

```
link-layer-overhead percent;
```

You can include this statement at the following hierarchy levels:

- `[edit interfaces` *`interface-name`* `mlfr-uni-nni-bundle-options]`
- `[edit interfaces` *`interface-name`* `unit` *`logical-unit-number`*`]`
- `[edit logical-systems` *`logical-system-name`* `interfaces` *`interface-name`* `unit` *`logical-unit-number`*`]`

You can configure the value to be from 0 percent through 50 percent.

SEE ALSO

Layer 2 Service Package Capabilities and Interfaces | **921**

Oversubscribing Interface Bandwidth on LSQ Interfaces | **860**

Configuring Guaranteed Minimum Rate on LSQ Interfaces | **866**

Link Services Configuration for Junos Interfaces | **847**

Enabling Inline LSQ Services

Inline Multilink PPP (*MLPPP*), Multilink Frame Relay (*FRF.16*), and Multilink Frame Relay End-to-End (*FRF.15*) for time-division multiplexing (*TDM*) WAN interfaces provide bundling services through the Packet Forwarding Engine without requiring a PIC or Dense Port Concentrator (*DPC*).

Traditionally, bundling services are used to bundle multiple low-speed links to create a higher bandwidth pipe. This combined bandwidth is available to traffic from all links and supports link fragmentation and interleaving (*LFI*) on the bundle, reducing high priority packet transmission delay.

This support includes multiple links on the same bundle as well as multiclass extension for MLPPP. Through this service you can enable bundling services without additional DPC slots to support Service DPC and free up the slots for other MICs.

The inline LSQ logical interface (referred to as lsq-) is a virtual service logical interface that resides on the Packet Forwarding Engine to provide Layer 2 bundling services that do not need a service PIC. The naming convention is `lsq-slot/pic/0`.

NOTE: Click here for a compatibility matrix of MICs currently supported by MPC1, MPC2, MPC3, MPC6, MPC8, and MPC9 on MX240, MX480, MX960, MX2008, MX2010, MX2020, and MX10003 routers.

A Type1 MPC has only one logical unit (LU); therefore only one LSQ logical interface can be created. When configuring a Type1 MPC, use PIC slot 0. Type2 MPC has two LUs; therefore two LSQ logical interfaces can be created. When configuring a Type2 MPC, use PIC slot 0 and slot 2.

Configure each LSQ logical interface with one loopback stream. This stream can be shaped like a regular stream, and is shared with other inline interfaces, such as the inline services (*SI*) interface.

To support FRF.16 bundles, create logical interfaces with the naming convention `lsq-slot/pic/0:`*`bundle_id`*, where *bundle_id* can range from 0 to 254. You can configure logical interfaces created on the main LSQ logical interface as MLPPP or FRF.16.

Because SI and LSQ logical interfaces might share the same stream, and there could be multiple LSQ logical interfaces on that stream, any logical interface-related shaping is configured at the Layer 2 node instead of the Layer 1 node. As a result, when SI is enabled, instead of limiting the stream bandwidth to 1Gb or 10Gb based on the configuration, only the Layer 2 queue allocated for the SI interface is shaped at 1Gb or 10Gb.

For MLPPP and FRF.15, each LSQ logical interface is shaped based on the total bundle bandwidth (sum of member link bandwidths with control packet flow overhead) by configuring one unique Layer 3 node per bundle. Similarly, each FRF.16 logical interface is shaped based on total bundle bandwidth by configuring one unique Layer 2 node per bundle. FRF16 logical interface data-link connection identifiers (*DLCI*s) are mapped to Layer 3 nodes.

To enable inline LSQ services and create the `lsq-` logical interface for the specified PIC, specify the *multi-link-layer-2-inline* and *mlfr-uni-nni-bundles-inline* configuration statements.

```
[edit chassis fpc number pic number]
user@host# set multi-link-layer-2-inline
user@host# set mlfr-uni-nni-bundles-inline number
```

NOTE: On MX80 and MX104 routers that have a single Packet Forwarding Engine, you can configure the LSQ logical interface only on FPC 0 and PIC 0. The channelized card must be in slot FPC 0/0 for the corresponding bundle to work.

For example, to enable inline service for PIC 0 on a Type1 MPC on slot 1:

```
[edit chassis fpc 1 pic 0]
user@host# set multi-link-layer-2-inline
user@host# set mlfr-uni-nni-bundles-inline 1
```

As a result, logical interfaces lsq-1/0/0, and lsq-1/0/0:0 are created. The number of inline multilink frame relay user-to-network interface (UNI) and network-to-network interface (NNI) bundles is set to 1.

For example, to enable inline service for both PIC 0 and PIC 2 on Type2 MPC installed in slot 5:

```
[edit chassis fpc 5 pic 0]
user@host# set multi-link-layer-2-inline
user@host# set mlfr-uni-nni-bundles-inline 1

[edit chassis fpc 5 pic 2]
user@host# set multi-link-layer-2-inline
user@host# set mlfr-uni-nni-bundles-inline 1
```

As a result, logical interfaces lsq-5/0/0, lsq-5/0/0:0, lsq-5/0/0:1, lsq-5/2/0, lsq-5/2/0:0, and lsq-5/2/0:1 are created. The number of inline multilink frame relay user-to-network interface (UNI) and network-to-network interface (NNI) bundles is set to 1.

NOTE: The PIC number here is only used as an anchor to choose the correct LU to bind the inline LSQ interface. The bundling services are operational as long as the Packet Forwarding Engine to which it is bound is operational, even if the logical PIC is offline.

SEE ALSO

https://www.juniper.net/documentation/en_US/junos/information-products/pathway-pages/services-interfaces/link-multilink-properties.html

mlfr-uni-nni-bundles-inline

multi-link-layer-2-inline

Configuring LSQ Interfaces as NxT1 or NxE1 Bundles Using MLPPP

IN THIS SECTION

● Example: Configuring an LSQ Interface as an NxT1 Bundle Using MLPPP | **949**

To configure an *N*xT1 bundle using MLPPP, you aggregate *N* different T1 links into a bundle. The *N*xT1 bundle is called a logical interface, because it can represent, for example, a routing adjacency. To aggregate T1 links into a an MLPPP bundle, include the `bundle` statement at the `[edit interfaces t1-`*`fpc/pic/port`*` unit `*`logical-unit-number`*` family mlppp]` hierarchy level:

```
[edit interfaces t1-fpc/pic/port unit logical-unit-number family mlppp]
bundle lsq-fpc/pic/port.logical-unit-number;
```

NOTE: Link services IQ interfaces support both T1 and E1 physical interfaces. These instructions apply to T1 interfaces, but the configuration for E1 interfaces is similar.

To configure the link services IQ interface properties, include the following statements at the `[edit interfaces lsq-`*`fpc/pic/port`*` unit `*`logical-unit-number`*`]` hierarchy level:

```
[edit interfaces lsq-fpc/pic/port unit logical-unit-number]
drop-timeout milliseconds;
encapsulation multilink-ppp;
fragment-threshold bytes;
link-layer-overhead percent;
minimum-links number;
mrru bytes;
short-sequence;
family inet {
    address address;
}
```

NOTE: ACX Series routers do not support drop-timeout and link-layer-overhead properties.

The logical link services IQ interface represents the MLPPP bundle. For the MLPPP bundle, there are four associated queues on M Series routers and eight associated queues on M320 and T Series routers. A scheduler removes packets from the queues according to a scheduling policy. Typically, you designate one queue to have strict priority, and the remaining queues are serviced in proportion to weights you configure.

For MLPPP, assign a single scheduler map to the link services IQ interface (`lsq`) and to each constituent link. The default schedulers for M Series and T Series routers, which assign 95, 0, 0, and 5 percent bandwidth for the transmission rate and buffer size of queues 0, 1, 2, and 3, are not adequate when you configure LFI or multiclass traffic. Therefore, for MLPPP, you should configure a single scheduler with nonzero percent transmission rates and buffer sizes for queues 0 through 3, and assign this scheduler to the link services IQ interface (`lsq`) and to each constituent link, as shown in "Example: Configuring an LSQ Interface as an NxT1 Bundle Using MLPPP" on page 949 .

NOTE: For M320 and T Series routers, the default scheduler transmission rate and buffer size percentages for queues 0 through 7 are 95, 0, 0, 5, 0, 0, 0, and 0 percent.

If the member link belonging to one MLPP, MLFR, or MFR bundle interface is moved to another bundle interface, or the links are swapped between two bundle interfaces, a commit is required between the delete and add operations to ensure that the configuration is applied correctly.

If the bundle has more than one link, you must include the `per-unit-scheduler` statement at the `[edit interfaces lsq-`*`fpc/pic/port`*`]` hierarchy level:

```
[edit interfaces lsq-fpc/pic/port]
per-unit-scheduler;
```

To configure and apply the scheduling policy, include the following statements at the `[edit class-of-service]` hierarchy level:

```
[edit class-of-service]
interfaces {
    t1-fpc/pic/port unit logical-unit-number {
        scheduler-map map-name;
    }
}
forwarding-classes {
    queue queue-number class-name;
}
scheduler-maps {
    map-name {
        forwarding-class class-name scheduler scheduler-name;
    }
}
schedulers {
    scheduler-name {
        buffer-size (percent percentage | remainder | temporal microseconds);
        priority priority-level;
        transmit-rate (rate | percent percentage | remainder) <exact>;
    }
}
```

For link services IQ interfaces, a strict-high-priority queue might starve the other three queues because traffic in a strict-high priority queue is transmitted before any other queue is serviced. This implementation is unlike the standard Junos CoS implementation in which a strict-high-priority queue does round-robin with high-priority queues, as described in the Class of Service User Guide (Routers and EX9200 Switches).

After the scheduler removes a packet from a queue, a certain action is taken. The action depends on whether the packet came from a multilink encapsulated queue (fragmented and sequenced) or a nonencapsulated queue (hashed with no fragmentation). Each queue can be designated as either multilink encapsulated or nonencapsulated, independently of the other. By default, traffic in all forwarding classes is multilink encapsulated. To configure packet fragmentation handling on a queue, include the `fragmentation-maps` statement at the `[edit class-of-service]` hierarchy level:

```
fragmentation-maps {
    map-name {
        forwarding-class class-name {
            fragment-threshold bytes;
            multilink-class number;
            no-fragmentation;
        }
    }
}
```

For *N*xT1 bundles using MLPPP, the byte-wise load balancing used in multilink-encapsulated queues is superior to the flow-wise load balancing used in nonencapsulated queues. All other considerations are equal. Therefore, we recommend that you configure all queues to be multilink encapsulated. You do this by including the `fragment-threshold` statement in the configuration. If you choose to set traffic on a queue to be nonencapsulated rather than multilink encapsulated, include the `no-fragmentation` statement in the fragmentation map. You use the `multilink-class` statement to map a forwarding class into a multiclass MLPPP (MCML). . For more information about fragmentation maps, see "Configuring CoS Fragmentation by Forwarding Class on LSQ Interfaces" on page 853 .

When a packet is removed from a multilink-encapsulated queue, the software gives the packet an MLPPP header. The MLPPP header contains a sequence number field, which is filled with the next available sequence number from a counter. The software then places the packet on one of the *N* different T1 links. The link is chosen on a packet-by-packet basis to balance the load across the various T1 links.

If the packet exceeds the minimum link MTU, or if a queue has a fragment threshold configured at the `[edit class-of-service fragmentation-maps` *`map-name`* `forwarding-class` *`class-name`*`]` hierarchy level, the software splits the packet into two or more fragments, which are assigned consecutive multilink sequence numbers. The outgoing link for each fragment is selected independently of all other fragments.

If you do not include the `fragment-threshold` statement in the fragmentation map, the fragmentation threshold you set at the `[edit interfaces` *`interface-name`* `unit` *`logical-unit-number`*`]` hierarchy level is the default for all forwarding classes. If you do not set a maximum fragment size anywhere in the configuration, packets are fragmented if they exceed the smallest MTU of all the links in the bundle.

Even if you do not set a maximum fragment size anywhere in the configuration, you can configure the maximum received reconstructed unit (MRRU) by including the `mrru` statement at the `[edit interfaces lsq-`*`fpc/pic/port`* `unit` *`logical-unit-number`*`]` hierarchy level. The MRRU is similar to the MTU, but is specific to link services interfaces. By default the MRRU size is 1500 bytes, and you can configure it to be from

1500 through 4500 bytes. For more information, see *Configuring MRRU on Multilink and Link Services Logical Interfaces*.

When a packet is removed from a nonencapsulated queue, it is transmitted with a plain PPP header. Because there is no MLPPP header, there is no sequence number information. Therefore, the software must take special measures to avoid packet reordering. To avoid packet reordering, the software places the packet on one of the *N* different T1 links. The link is determined by hashing the values in the header. For IP, the software computes the hash based on source address, destination address, and IP protocol. For MPLS, the software computes the hash based on up to five MPLS labels, or four MPLS labels and the IP header.

For UDP and TCP the software computes the hash based on the source and destination ports, as well as source and destination IP addresses. This guarantees that all packets belonging to the same TCP/UDP flow always pass through the same T1 link, and therefore cannot be reordered. However, it does not guarantee that the load on the various T1 links is balanced. If there are many flows, the load is usually balanced.

The *N* different T1 interfaces link to another router, which can be from Juniper Networks or another vendor. The router at the far end gathers packets from all the T1 links. If a packet has an MLPPP header, the sequence number field is used to put the packet back into sequence number order. If the packet has a plain PPP header, the software accepts the packet in the order in which it arrives and makes no attempt to reassemble or reorder the packet.

Example: Configuring an LSQ Interface as an NxT1 Bundle Using MLPPP

```
[edit chassis]
fpc 1 {
    pic 3 {
        adaptive-services {
            service-package layer-2;
        }
    }
}
[edit interfaces]
t1-0/0/0 {
    encapsulation ppp;
    unit 0 {
        family mlppp {
            bundle lsq-1/3/0.1; # This adds t1-0/0/0 to the specified bundle.
        }
    }
}
t1-0/0/1 {
    encapsulation ppp;
    unit 0 {
        family mlppp {
            bundle lsq-1/3/0.1;
        }
    }
}
lsq-1/3/0 {
    unit 1 { # This is the virtual link that concatenates multiple T1s.
    encapsulation multilink-ppp;
    drop-timeout 1000;
    fragment-threshold 128;
    link-layer-overhead 0.5;
    minimum-links 2;
    mrru 4500;
    short-sequence;
    family inet {
        address 10.2.3.4/24;
    }
}
[edit interfaces]
lsq-1/3/0 {
    per-unit-scheduler;
}
[edit class-of-service]
interfaces {
    lsq-1/3/0 { # multilink PPP constituent link
    unit 0 {
        scheduler-map sched-map1;
    }
}
t1-0/0/0 { # multilink PPP constituent link
unit 0 {
    scheduler-map sched-map1;
}
t1-0/0/1 { # multilink PPP constituent link
unit 0 {
    scheduler-map sched-map1;
}
forwarding-classes {
    queue 0 be;
    queue 1 ef;
    queue 2 af;
    queue 3 nc;
}
scheduler-maps {
    sched-map1 {
        forwarding-class af scheduler af-scheduler;
        forwarding-class be scheduler be-scheduler;
        forwarding-class ef scheduler ef-scheduler;
        forwarding-class nc scheduler nc-scheduler;
    }
}
schedulers {
    af-scheduler {
        transmit-rate percent 30;
        buffer-size percent 30;
        priority low;
    }
    be-scheduler {
        transmit-rate percent 25;
        buffer-size percent 25;
        priority low;
    }
    ef-scheduler {
        transmit-rate percent 40;
        buffer-size percent 40;
        priority strict-high;    # voice queue
    }
    nc-scheduler {
        transmit-rate percent 5;
        buffer-size percent 5;
        priority high;
    }
}
fragmentation-maps {
    fragmap-1 {
        forwarding-class be {
            fragment-threshold 180;
        }
        forwarding-class ef {
            fragment-threshold 100;
        }
    }
}
[edit interfaces]
```

```
lsq-1/3/0 {
    unit 0 {
        fragmentation-map fragmap-1;
    }
}
```

SEE ALSO

Layer 2 Service Package Capabilities and Interfaces | 921

Link Services Configuration for Junos Interfaces | 847

Configuring LSQ Interfaces as NxT1 or NxE1 Bundles Using FRF.16

IN THIS SECTION

- Example: Configuring an LSQ Interface as an NxT1 Bundle Using FRF.16 | 956

To configure an *N*xT1 bundle using FRF.16, you aggregate *N* different T1 links into a bundle. The *N*xT1 bundle carries a potentially large number of Frame Relay PVCs, identified by their DLCIs. Each DLCI is called a logical interface, because it can represent, for example, a routing adjacency.

To aggregate T1 links into an FRF.16 bundle, include the `mlfr-uni-nni-bundles` statement at the `[edit chassis fpc` *`slot-number`* `pic` *`slot-number`*`]` hierarchy level and include the `bundle` statement at the `[edit interfaces t1-`*`fpc/pic/port`* `unit` *`logical-unit-number`* `family mlfr-uni-nni]` hierarchy level:

```
[edit chassis fpc slot-number pic slot-number]
mlfr-uni-nni-bundles number;

[edit interfaces t1-fpc/pic/port unit logical-unit-number family mlfr-uni-nni]
bundle lsq-fpc/pic/port:channel;
```

NOTE: Link services IQ interfaces support both T1 and E1 physical interfaces. These instructions apply to T1 interfaces, but the configuration for E1 interfaces is similar.

To configure the link services IQ interface properties, include the following statements at the `[edit interfaces lsq- fpc/pic/port:channel]` hierarchy level:

```
[edit interfaces lsq- fpc/pic/port:channel]
encapsulation multilink-frame-relay-uni-nni;
dce;
mlfr-uni-nni-options {
    acknowledge-retries number;
    acknowledge-timer milliseconds;
    action-red-differential-delay (disable-tx | remove-link);
    drop-timeout milliseconds;
    fragment-threshold bytes;
    hello-timer milliseconds;
    link-layer-overhead percent;
    lmi-type (ansi | itu);
    minimum-links number;
    mrru bytes;
    n391 number;
    n392 number;
    n393 number;
    red-differential-delay milliseconds;
    t391 number;
    t392 number;
    yellow-differential-delay milliseconds;
}
unit logical-unit-number {
    dlci dlci-identifier;
    family inet {
        address address;
    }
}
```

The link services IQ channel represents the FRF.16 bundle. Four queues are associated with each DLCI. A scheduler removes packets from the queues according to a scheduling policy. On the link services IQ interface, you typically designate one queue to have strict priority. The remaining queues are serviced in proportion to weights you configure.

For link services IQ interfaces, a strict-high-priority queue might starve the other three queues because traffic in a strict-high-priority queue is transmitted before any other queue is serviced. This implementation is unlike the standard Junos CoS implementation in which a strict-high-priority queue does round-robin with high-priority queues, as described in the Class of Service User Guide (Routers and EX9200 Switches).

If the bundle has more than one link, you must include the `per-unit-scheduler` statement at the `[edit interfaces lsq-`*`fpc/pic/port:channel`*`]` hierarchy level:

```
[edit interfaces lsq-fpc/pic/port:channel]
per-unit-scheduler;
```

For FRF.16, you can assign a single scheduler map to the link services IQ interface (`lsq`) and to each link services IQ DLCI, or you can assign different scheduler maps to the various DLCIs of the bundle, as shown in "Example: Configuring an LSQ Interface as an NxT1 Bundle Using FRF.16" on page 956 .

For the constituent links of an FRF.16 bundle, you do not need to configure a custom scheduler. Because LFI and multiclass are not supported for FRF.16, the traffic from each constituent link is transmitted from queue 0. This means you should allow most of the bandwidth to be used by queue 0. For M Series and T Series routers, the default schedulers' transmission rate and buffer size percentages for queues 0 through 3 are 95, 0, 0, and 5 percent. These default schedulers send all user traffic to queue 0 and all network-control traffic to queue 3, and therefore are well suited to the behavior of FRF.16. If desired, you can configure a custom scheduler that explicitly replicates the 95, 0, 0, and 5 percent queuing behavior, and apply it to the constituent links.

NOTE: For M320 and T Series routers, the default scheduler transmission rate and buffer size percentages for queues 0 through 7 are 95, 0, 0, 5, 0, 0, 0, and 0 percent.

If the member link belonging to one MLPP, MLFR, or MFR bundle interface is moved to another bundle interface, or the links are swapped between two bundle interfaces, a commit is required between the delete and add operations to ensure that the configuration is applied correctly.

To configure and apply the scheduling policy, include the following statements at the `[edit class-of-service]` hierarchy level:

```
[edit class-of-service]
interfaces {
    lsq-fpc/pic/port:channel {
        unit logical-unit-number {
            scheduler-map map-name;
        }
    }
}
forwarding-classes {
    queue queue-number class-name;
}
scheduler-maps {
    map-name {
        forwarding-class class-name scheduler scheduler-name;
    }
}
schedulers {
    scheduler-name {
        buffer-size (percent percentage | remainder | temporal microseconds);
        priority priority-level;
        transmit-rate (rate | percent percentage | remainder) <exact>;
    }
}
```

To configure packet fragmentation handling on a queue, include the `fragmentation-maps` statement at the `[edit class-of-service]` hierarchy level:

```
[edit class-of-service]
fragmentation-maps {
    map-name {
        forwarding-class class-name {
            fragment-threshold bytes;
        }
    }
}
```

For FRF.16 traffic, only multilink encapsulated (fragmented and sequenced) queues are supported. This is the default queuing behavior for all forwarding classes. FRF.16 does not allow for nonencapsulated traffic because the protocol requires that all packets carry the fragmentation header. If a large packet is split into multiple fragments, the fragments must have consecutive sequential numbers. Therefore, you cannot include the `no-fragmentation` statement at the `[edit class-of-service fragmentation-maps` *`map-name`* `forwarding-class` *`class-name`*`]` hierarchy level for FRF.16 traffic. For FRF.16, if you want to carry voice or any other latency-sensitive traffic, you should not use slow links. At T1 speeds and above, the serialization delay is small enough so that you do not need to use explicit LFI.

When a packet is removed from a multilink-encapsulated queue, the software gives the packet an FRF.16 header. The FRF.16 header contains a sequence number field, which is filled with the next available sequence number from a counter. The software then places the packet on one of the *N* different T1 links. The link is chosen on a packet-by-packet basis to balance the load across the various T1 links.

If the packet exceeds the minimum link MTU, or if a queue has a fragment threshold configured at the `[edit class-of-service fragmentation-maps` *`map-name`* `forwarding-class` *`class-name`*`]` hierarchy level, the software splits the packet into two or more fragments, which are assigned consecutive multilink sequence numbers. The outgoing link for each fragment is selected independently of all other fragments.

If you do not include the `fragment-threshold` statement in the fragmentation map, the fragmentation threshold you set at the `[edit interfaces` *`interface-name`* `unit` *`logical-unit-number`*`]` or `[edit interfaces` *`interface-name`* `mlfr-uni-nni-bundle-options]` hierarchy level is the default for all forwarding classes. If you do not set a maximum fragment size anywhere in the configuration, packets are fragmented if they exceed the smallest MTU of all the links in the bundle.

Even if you do not set a maximum fragment size anywhere in the configuration, you can configure the maximum received reconstructed unit (MRRU) by including the `mrru` statement at the `[edit interfaces lsq-`*`fpc/pic/port`* `unit` *`logical-unit-number`*`]` or `[edit interfaces` *`interface-name`* `mlfr-uni-nni-bundle-options]` hierarchy level. The MRRU is similar to the MTU but is specific to link services interfaces. By default, the MRRU size is 1500 bytes, and you can configure it to be from 1500 through 4500 bytes. For more information, see *Configuring MRRU on Multilink and Link Services Logical Interfaces*.

The *N* different T1 interfaces link to another router, which can be from Juniper Networks or another vendor. The router at the far end gathers packets from all the T1 links. Because each packet has an FRF.16 header, the sequence number field is used to put the packet back into sequence number order.

Example: Configuring an LSQ Interface as an NxT1 Bundle Using FRF.16

Configure an *N*xT1 bundle using FRF.16 with multiple CoS scheduler maps:

```
[edit chassis fpc 1 pic 3]
adaptive-services {
    service-package layer-2;
}
mlfr-uni-nni-bundles 2; # Creates channelized LSQ interfaces/FRF.16 bundles.
[edit interfaces]
t1-0/0/0 {
    encapsulation multilink-frame-relay-uni-nni;
    unit 0 {
        family mlfr-uni-nni {
            bundle lsq-1/3/0:1;
        }
    }
}
t1-0/0/1 {
    encapsulation multilink-frame-relay-uni-nni;
    unit 0 {
        family mlfr-uni-nni {
            bundle lsq-1/3/0:1;
        }
    }
}
```

```
lsq-1/3/0:1 { # Bundle link consisting of t1-0/0/0 and t1-0/0/1
per-unit-scheduler;
encapsulation multilink-frame-relay-uni-nni;
dce; # One end needs to be configured as DCE.
mlfr-uni-nni-bundle-options {
    drop-timeout 180;
    fragment-threshold 64;
    hello-timer 180;
    minimum-links 2;
    mrru 3000;
    link-layer-overhead 0.5;
}
unit 0 {
    dlci 26; # Each logical unit maps a single DLCI.
    family inet {
        address 10.2.3.4/24;
    }
}
unit 1 {
    dlci 42;
    family inet {
        address 10.20.30.40/24;
    }
}
unit 2 {
    dlci 69;
    family inet {
        address 10.20.30.40/24;
    }
}
[edit class-of-service]
scheduler-maps {
    sched-map-lsq0 {
        forwarding-class af scheduler af-scheduler-lsq0;
        forwarding-class be scheduler be-scheduler-lsq0;
        forwarding-class ef scheduler ef-scheduler-lsq0;
        forwarding-class nc scheduler nc-scheduler-lsq0;
    }
    sched-map-lsq1 {
        forwarding-class af scheduler af-scheduler-lsq1;
        forwarding-class be scheduler be-scheduler-lsq1;
        forwarding-class ef scheduler ef-scheduler-lsq1;
        forwarding-class nc scheduler nc-scheduler-lsq1;
    }
}
schedulers {
    af-scheduler-lsq0 {
        transmit-rate percent 60;
        buffer-size percent 60;
        priority low;
    }
    be-scheduler-lsq0 {
        transmit-rate percent 30;
        buffer-size percent 30;
        priority low;
    }
    ef-scheduler-lsq0 {
        transmit-rate percent 5;
        buffer-size percent 5;
        priority strict-high;
    }
    nc-scheduler-lsq0 {
        transmit-rate percent 5;
        buffer-size percent 5;
        priority high;
    }
    af-scheduler-lsq1 {
        transmit-rate percent 50;
        buffer-size percent 50;
        priority low;
    }
    be-scheduler-lsq1 {
        transmit-rate percent 30;
        buffer-size percent 30;
        priority low;
    }
    ef-scheduler-lsq1 {
        transmit-rate percent 15;
        buffer-size percent 15;
        priority strict-high;
    }
    nc-scheduler-lsq1 {
        transmit-rate percent 5;
        buffer-size percent 5;
        priority high;
    }
```

```
}
interfaces {
    lsq-1/3/0:1 { # MLFR FRF.16
    unit 0 {
        scheduler-map sched-map-lsq0;
    }
    unit 1 {
        scheduler-map sched-map-lsq1;
    }
}
```

SEE ALSO

Layer 2 Service Package Capabilities and Interfaces | 921

Inline Multilink Services | 939

Inline Multilink Services | 939

Link Services Configuration for Junos Interfaces | 847

Configuring LSQ Interfaces as NxT1 or NxE1 Bundles Using FRF.15

This example configures an *N*xT1 bundle using FRF.15 on a link services IQ interface. FRF.15 is similar to FRF.12, as described in "Configuring LSQ Interfaces for Single Fractional T1 or E1 Interfaces Using FRF.12" on page 939 . The difference is that FRF.15 supports multiple physical links in a bundle, whereas FRF.12 supports only one physical link per bundle. For the Junos OS implementation of FRF.15, you can configure one DLCI per physical link.

NOTE: Link services IQ interfaces support both T1 and E1 physical interfaces. This example refers to T1 interfaces, but the configuration for E1 interfaces is similar.

```
[edit interfaces]
lsq-1/3/0 {
    per-unit-scheduler;
    unit 0 {
        dlci 69;
        encapsulation multilink-frame-relay-end-to-end;
    }
}
unit 1 {
    dlci 13;
    encapsulation multilink-frame-relay-end-to-end;
}
# First physical link
t1-1/1/0:1 {
    encapsulation frame-relay;
    unit 0 {
        family mlfr-end-to-end {
            bundle lsq-1/3/0.0;
        }
    }
}
# Second physical link
t1-1/1/0:2 {
    encapsulation frame-relay;
    unit 0 {
        family mlfr-end-to-end {
            bundle lsq-1/3/0.0;
        }
    }
}
```

SEE ALSO

Layer 2 Service Package Capabilities and Interfaces | 921

Link Services Configuration for Junos Interfaces | 847

Configuring LSQ Interfaces for Single Fractional T1 or E1 Interfaces Using MLPPP and LFI

IN THIS SECTION

- Example: Configuring an LSQ Interface for a Fractional T1 Interface Using MLPPP and LFI | 964

When you configure a single fractional T1 interface, it is called a logical interface, because it can represent, for example, a routing adjacency.

The logical link services IQ interface represents the MLPPP bundle. Four queues are associated with the logical interface. A scheduler removes packets from the queues according to a scheduling policy. Typically, you designate one queue to have strict priority, and the remaining queues are serviced in proportion to weights you configure.

To configure a single fractional T1 interface using MLPPP and LFI, you associate one DS0 (fractional T1) interface with a link services IQ interface. To associate a fractional T1 interface with a link services IQ interface, include the `bundle` statement at the `[edit interfaces ds-`*`fpc/pic/port:channel`*` unit `*`logical-unit-number`*` family mlppp]` hierarchy level:

```
[edit interfaces ds-fpc/pic/port:channel unit logical-unit-number family mlppp]
bundle lsq-fpc/pic/port.logical-unit-number;
```

NOTE: Link services IQ interfaces support both T1 and E1 physical interfaces. These instructions apply to T1 interfaces, but the configuration for E1 interfaces is similar.

To configure the link services IQ interface properties, include the following statements at the `[edit interfaces lsq-`*`fpc/pic/port`*` unit `*`logical-unit-number`*`]` hierarchy level:

```
[edit interfaces lsq-fpc/pic/port unit logical-unit-number]
drop-timeout milliseconds;
encapsulation multilink-ppp;
fragment-threshold bytes;
link-layer-overhead percent;
minimum-links number;
mrru bytes;
short-sequence;
family inet {
    address address;
}
```

For MLPPP, assign a single scheduler map to the link services IQ (`lsq`) interface and to each constituent link. The default schedulers for M Series and T Series routers, which assign 95, 0, 0, and 5 percent bandwidth for the transmission rate and buffer size of queues 0, 1, 2, and 3, are not adequate when you configure LFI or multiclass traffic. Therefore, for MLPPP, you should configure a single scheduler with nonzero percent transmission rates and buffer sizes for queues 0 through 3, and assign this scheduler to the link services IQ (`lsq`) interface and to each constituent link and to each constituent link, as shown in "Example: Configuring an LSQ Interface for a Fractional T1 Interface Using MLPPP and LFI" on page 964 .

NOTE: For M320 and T Series routers, the default scheduler transmission rate and buffer size percentages for queues 0 through 7 are 95, 0, 0, 5, 0, 0, 0, and 0 percent.

To configure and apply the scheduling policy, include the following statements at the `[edit class-of-service]` hierarchy level:

```
[edit class-of-service]
interfaces {
    ds-fpc/pic/port.channel {
        scheduler-map map-name;
    }
}
forwarding-classes {
    queue queue-number class-name;
}
scheduler-maps {
    map-name {
        forwarding-class class-name scheduler scheduler-name;
    }
}
schedulers {
    scheduler-name {
        buffer-size (percent percentage | remainder | temporal microseconds);
        priority priority-level;
        transmit-rate (rate | percent percentage | remainder) <exact>;
    }
}
```

For link services IQ interfaces, a strict-high-priority queue might starve all the other queues because traffic in a strict-high priority queue is transmitted before any other queue is serviced. This implementation is unlike the standard Junos CoS implementation in which a strict-high-priority queue receives infinite credits and does round-robin with high-priority queues, as described in the Class of Service User Guide (Routers and EX9200 Switches).

After the scheduler removes a packet from a queue, a certain action is taken. The action depends on whether the packet came from a multilink encapsulated queue (fragmented and sequenced) or a nonencapsulated queue (hashed with no fragmentation). Each queue can be designated as either multilink encapsulated or nonencapsulated, independently of the other. By default, traffic in all forwarding classes is multilink encapsulated. To configure packet fragmentation handling on a queue, include the fragmentation-maps statement at the [edit class-of-service] hierarchy level:

```
[edit class-of-service]
fragmentation-maps {
    map-name {
        forwarding-class class-name {
            fragment-threshold bytes;
            no-fragmentation;
        }
    }
}
```

If you require the queue to transmit small packets with low latency, configure the queue to be nonencapsulated by including the no-fragmentation statement. If you require the queue to transmit large packets with normal latency, configure the queue to be multilink encapsulated by including the fragment-threshold statement. If you require the queue to transmit large packets with low latency, we recommend using a faster link and configuring the queue to be nonencapsulated. For more information about fragmentation maps, see "Configuring CoS Fragmentation by Forwarding Class on LSQ Interfaces" on page 853 .

When a packet is removed from a multilink-encapsulated queue, it is fragmented. If the packet exceeds the minimum link MTU, or if a queue has a fragment threshold configured at the [edit class-of-service fragmentation-maps *map-name* forwarding-class *class-name*] hierarchy level, the software splits the packet into two or more fragments, which are assigned consecutive multilink sequence numbers.

If you do not include the fragment-threshold statement in the fragmentation map, the fragmentation threshold you set at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level is the default for all forwarding classes. If you do not set a maximum fragment size anywhere in the configuration, packets are fragmented if they exceed the smallest MTU of all the links in the bundle.

Even if you do not set a maximum fragment size anywhere in the configuration, you can configure the maximum received reconstructed unit (MRRU) by including the mrru statement at the [edit interfaces lsq-*fpc*/*pic*/*port* unit *logical-unit-number*] hierarchy level. The MRRU is similar to the MTU, but is specific to link services interfaces. By default the MRRU size is 1500 bytes, and you can configure it to be from 1500 through 4500 bytes. For more information, see *Configuring MRRU on Multilink and Link Services Logical Interfaces*.

When a packet is removed from a multilink-encapsulated queue, the software gives the packet an MLPPP header. The MLPPP header contains a sequence number field, which is filled with the next available sequence number from a counter. The software then places the packet on the fractional T1 link. Traffic from another queue might be interleaved between two fragments of the packet.

When a packet is removed from a nonencapsulated queue, it is transmitted with a plain PPP header. The packet is then placed on the fractional T1 link as soon as possible. If necessary, the packet is placed between the fragments of a packet from another queue.

The fractional T1 interface links to another router, which can be from Juniper Networks or another vendor. The router at the far end gathers packets from the fractional T1 link. If a packet has an MLPPP header, the software assumes the packet is a fragment of a larger packet, and the fragment number field is used to reassemble the larger packet. If the packet has a plain PPP header, the software accepts the packet in the order in which it arrives, and the software makes no attempt to reassemble or reorder the packet.

Example: Configuring an LSQ Interface for a Fractional T1 Interface Using MLPPP and LFI

Configure a single fractional T1 logical interface:

```
[edit interfaces]
lsq-0/2/0 {
    per-unit-scheduler;
    unit 0 {
        encapsulation multilink-ppp;
        link-layer-overhead 0.5;
        family inet {
            address 10.40.1.1/30;
        }
    }
}
ct3-1/0/0 {
    partition 1 interface-type ct1;
}
ct1-1/0/0:1 {
    partition 1 timeslots 1-2 interface-type ds;
}
ds-1/0/0:1:1 {
    encapsulation ppp;
    unit 0 {
        family mlppp {
            bundle lsq-0/2/0.0;
        }
    }
}
[edit class-of-service]
interfaces {
    ds-1/0/0:1:1 { # multilink PPP constituent link
    unit 0 {
        scheduler-map sched-map1;
    }
}
forwarding-classes {
    queue 0 be;
    queue 1 ef;
    queue 2 af;
    queue 3 nc;
}
scheduler-maps {
    sched-map1 {
        forwarding-class af scheduler af-scheduler;
        forwarding-class be scheduler be-scheduler;
        forwarding-class ef scheduler ef-scheduler;
        forwarding-class nc scheduler nc-scheduler;
    }
}
schedulers {
    af-scheduler {
        transmit-rate percent 20;
        buffer-size percent 20;
        priority low;
    }
    be-scheduler {
        transmit-rate percent 20;
        buffer-size percent 20;
        priority low;
    }
    ef-scheduler {
        transmit-rate percent 50;
        buffer-size percent 50;
        priority strict-high;    # voice queue
    }
    nc-scheduler {
        transmit-rate percent 10;
        buffer-size percent 10;
        priority high;
    }
}
fragmentation-maps {
    fragmap-1 {
        forwarding-class be {
            fragment-threshold 180;
        }
        forwarding-class ef {
            fragment-threshold 100;
        }
    }
}
[edit interfaces]
lsq-0/2/0 {
    unit 0 {
        fragmentation-map fragmap-1;
    }
}
```

SEE ALSO

Layer 2 Service Package Capabilities and Interfaces | 921

Link Services Configuration for Junos Interfaces | 847

Configuring LSQ Interfaces for Single Fractional T1 or E1 Interfaces Using FRF.12

IN THIS SECTION

- Examples: Configuring an LSQ Interface for a Fractional T1 Interface Using FRF.12 | 970

To configure a single fractional T1 interface using FRF.16, you associate a DS0 interface with a link services IQ (`lsq`) interface. When you configure a single fractional T1, the fractional T1 carries a potentially large number of Frame Relay PVCs identified by their DLCIs. Each DLCI is called a logical interface, because it can represent, for example, a routing adjacency. To associate the DS0 interface with a link services IQ interface, include the `bundle` statement at the `[edit interfaces ds-`*`fpc/pic/port`*`:`*`channel`*` unit `*`logical-unit-number`*` family mlfr-end-to-end]` hierarchy level:

```
[edit interfaces ds-fpc/pic/port:channel unit logical-unit-number family mlfr-end-to-end]
bundle lsq-fpc/pic/port.logical-unit-number;
```

NOTE: Link services IQ interfaces support both T1 and E1 physical interfaces. These instructions apply to T1 interfaces, but the configuration for E1 interfaces is similar.

To configure the link services IQ interface properties, include the following statements at the `[edit interfaces lsq-`*`fpc/pic/port`* `unit` *`logical-unit-number`*`]` hierarchy level:

```
[edit interfaces lsq-fpc/pic/port unit logical-unit-number]
drop-timeout milliseconds;
encapsulation multilink-frame-relay-end-to-end;
fragment-threshold bytes;
link-layer-overhead percent;
minimum-links number;
mrru bytes;
short-sequence;
family inet {
    address address;
}
```

The logical link services IQ interface represents the FRF.12 bundle. Four queues are associated with each logical interface. A scheduler removes packets from the queues according to a scheduling policy. Typically, you designate one queue to have strict priority, and the remaining queues are serviced in proportion to weights you configure.

For FRF.12, assign a single scheduler map to the link services IQ interface (`lsq`) and to each constituent link. For M Series and T Series routers, the default schedulers, which assign 95, 0, 0, and 5 percent bandwidth for the transmission rate and buffer size of queues 0, 1, 2, and 3, are not adequate when you configure LFI or multiclass traffic. Therefore, for FRF.12, you should configure schedulers with nonzero percent transmission rates and buffer sizes for queues 0 through 3, and assign them to the link services IQ interface (`lsq`) and to each constituent link, as shown in "Examples: Configuring an LSQ Interface for a Fractional T1 Interface Using FRF.12" on page 970 .

NOTE: For M320 and T Series routers, the default scheduler transmission rate and buffer size percentages for queues 0 through 7 are 95, 0, 0, 5, 0, 0, 0, and 0 percent.

To configure and apply the scheduling policy, include the following statements at the `[edit class-of-service]` hierarchy level:

```
[edit class-of-service]
interfaces {
    ds-fpc/pic/port.channel {
        scheduler-map map-name;
    }
}
forwarding-classes {
    queue queue-number class-name;
}
scheduler-maps {
    map-name {
        forwarding-class class-name scheduler scheduler-name;
    }
}
schedulers {
    scheduler-name {
        buffer-size (percent percentage | remainder | temporal microseconds);
        priority priority-level;
        transmit-rate (rate | percent percentage | remainder) <exact>;
    }
}
```

For link services IQ interfaces, a strict-high-priority queue might starve the other three queues because traffic in a strict-high-priority queue is transmitted before any other queue is serviced. This implementation is unlike the standard Junos CoS implementation in which a strict-high-priority queue does round-robin with high-priority queues, as described in the Class of Service User Guide (Routers and EX9200 Switches).

After the scheduler removes a packet from a queue, a certain action is taken. The action depends on whether the packet came from a multilink encapsulated queue (fragmented and sequenced) or a nonencapsulated queue (hashed with no fragmentation). Each queue can be designated as either multilink encapsulated or nonencapsulated, independently of the other. By default, traffic in all forwarding classes is multilink encapsulated. To configure packet fragmentation handling on a queue, include the `fragmentation-maps` statement at the `[edit class-of-service]` hierarchy level:

```
[edit class-of-service]
fragmentation-maps {
    map-name {
        forwarding-class class-name {
            fragment-threshold bytes;
            no-fragmentation;
        }
    }
}
```

If you require the queue to transmit small packets with low latency, configure the queue to be nonencapsulated by including the `no-fragmentation` statement. If you require the queue to transmit large packets with normal latency, configure the queue to be multilink encapsulated by including the `fragment-threshold` statement. If you require the queue to transmit large packets with low latency, we recommend using a faster link and configuring the queue to be nonencapsulated. For more information about fragmentation maps, see "Configuring CoS Fragmentation by Forwarding Class on LSQ Interfaces" on page 853 .

When a packet is removed from a multilink-encapsulated queue, it is fragmented. If the packet exceeds the minimum link MTU, or if a queue has a fragment threshold configured at the `[edit class-of-service fragmentation-maps` *`map-name`* `forwarding-class` *`class-name`*`]` hierarchy level, the software splits the packet into two or more fragments, which are assigned consecutive multilink sequence numbers.

If you do not include the `fragment-threshold` statement in the fragmentation map, the fragmentation threshold you set at the `[edit interfaces` *`interface-name`* `unit` *`logical-unit-number`*`]` hierarchy level is the default for all forwarding classes. If you do not set a maximum fragment size anywhere in the configuration, packets are fragmented if they exceed the smallest MTU of all the links in the bundle.

Even if you do not set a maximum fragment size anywhere in the configuration, you can configure the maximum received reconstructed unit (MRRU) by including the `mrru` statement at the `[edit interfaces lsq-`*`fpc/pic/port`* `unit` *`logical-unit-number`*`]` hierarchy level. The MRRU is similar to the MTU but is specific to link services interfaces. By default, the MRRU size is 1500 bytes, and you can configure it to be from 1500 through 4500 bytes. For more information, see *Configuring MRRU on Multilink and Link Services Logical Interfaces*.

When a packet is removed from a multilink-encapsulated queue, the software gives the packet an FRF.12 header. The FRF.12 header contains a sequence number field, which is filled with the next available sequence number from a counter. The software then places the packet on the fractional T1 link. Traffic from another queue might be interleaved between two fragments of the packet.

When a packet is removed from a nonencapsulated queue, it is transmitted with a plain Frame Relay header. The packet is then placed on the fractional T1 link as soon as possible. If necessary, the packet is placed between the fragments of a packet from another queue.

The fractional T1 interface links to another router, which can be from Juniper Networks or another vendor. The router at the far end gathers packets from the fractional T1 link. If a packet has an FRF.12 header, the software assumes the packet is a fragment of a larger packet, and the fragment number field is used to reassemble the larger packet. If the packet has a plain Frame Relay header, the software accepts the packet in the order in which it arrives, and the software makes no attempt to reassemble or reorder the packet.

A whole packet from a nonencapsulated queue can be placed between fragments of a multilink-encapsulated queue. However, fragments from one multilink-encapsulated queue cannot be interleaved with fragments from another multilink-encapsulated queue. This is the intent of the specification FRF.12, *Frame Relay Fragmentation Implementation Agreement*. If fragments from two different queues were interleaved, the header fields might not have enough information to separate the fragments.

Examples: Configuring an LSQ Interface for a Fractional T1 Interface Using FRF.12

FRF.12 with Fragmentation and Without LFI

This example shows a 128 KB DS0 interface. There is one traffic stream on `ge-0/0/0`, which is classified into queue 0 (`be`). Packets are fragmented in the link services IQ (`lsq-`) interface according to the threshold configured in the fragmentation map.

```
[edit chassis]
fpc 0 {
    pic 3 {
        adaptive-services {
            service-package layer-2;
        }
    }
}
[edit interfaces]
ge-0/0/0 {
    unit 0 {
        family inet {
            address 20.1.1.1/24 {
                arp 20.1.1.2 mac 00.00.5e.00.53.56;
            }
        }
    }
}
ce1-0/2/0 {
    partition 1 timeslots 1-2 interface-type ds;
}
ds-0/2/0:1 {
    no-keepalives;
    dce;
    encapsulation frame-relay;
    unit 0 {
        dlci 100;
        family mlfr-end-to-end {
            bundle lsq-0/3/0.0;
        }
    }
}
lsq-0/3/0 {
    per-unit-scheduler;
    unit 0 {
        encapsulation multilink-frame-relay-end-to-end;
        family inet {
            address 10.200.0.78/30;
        }
    }
}
fxp0 {
    unit 0 {
        family inet {
            address 172.16.1.162/24;
        }
    }
}
lo0 {
    unit 0 {
        family inet {
            address 10.0.0.1/32;
        }
    }
}
[edit class-of-service]
forwarding-classes {
    queue 0 be;
    queue 1 ef;
    queue 2 af;
    queue 3 nc;
}
interfaces {
    lsq-0/3/0 {
        unit 0 {
            fragmentation-map map1;
        }
    }
}
fragmentation-maps {
    map1 {
        forwarding-class {
            be {
                fragment-threshold 160;
            }
        }
    }
}
```

FRF.12 with Fragmentation and LFI

This example shows a 512 KB DS0 bundle and four traffic streams on `ge-0/0/0` that are classified into four queues. The fragment size is 160 for queue 0, queue 1, and queue 2. The voice stream on queue 3 has LFI configured.

```
[edit chassis]
fpc 0 {
    pic 3 {
        adaptive-services {
            service-package layer-2;
        }
    }
}
[edit interfaces]
ge-0/0/0 {
    unit 0 {
        family inet {
            address 20.1.1.1/24 {
                arp 20.1.1.2 mac 00.00.5e.00.53.56;
            }
        }
    }
    ce1-0/2/0 {
        partition 1 timeslots 1-8 interface-type ds;
    }
    ds-0/2/0:1 {
        no-keepalives;
        dce;
        encapsulation frame-relay;
        unit 0 {
            dlci 100;
            family mlfr-end-to-end {
                bundle lsq-0/3/0.0;
            }
        }
    }
    lsq-0/3/0 {
        per-unit-scheduler;
        unit 0 {
            encapsulation multilink-frame-relay-end-to-end;
            family inet {
                address 10.200.0.78/30;
            }
        }
    }
    [edit class-of-service]
    classifiers {
        inet-precedence ge-interface-classifier {
            forwarding-class be {
                loss-priority low code-points 000;
            }
            forwarding-class ef {
                loss-priority low code-points 010;
            }
            forwarding-class af {
                loss-priority low code-points 100;
            }
            forwarding-class nc {
                loss-priority low code-points 110;
            }
        }
    }
    forwarding-classes {
        queue 0 be;
        queue 1 ef;
        queue 2 af;
        queue 3 nc;
    }
    interfaces {
        lsq-0/3/0 {
            unit 0 {
                scheduler-map sched2;
                fragmentation-map map2;
            }
        }
        ds-0/2/0:1 {
            scheduler-map link-map2;
        }
        ge-0/0/0 {
            unit 0 {
                classifiers {
                    inet-precedence ge-interface-classifier;
                }
            }
        }
    }
    scheduler-maps {
        sched2 {
            forwarding-class be scheduler economy;
            forwarding-class ef scheduler business;
            forwarding-class af scheduler stream;
            forwarding-class nc scheduler voice;
        }
        link-map2 {
            forwarding-class be scheduler link-economy;
            forwarding-class ef scheduler link-business;
            forwarding-class af scheduler link-stream;
            forwarding-class nc scheduler link-voice;
        }
    }
    fragmentation-maps {
        map2 {
            forwarding-class {
                be {
                    fragment-threshold 160;
                }
                ef {
                    fragment-threshold 160;
                }
                af {
                    fragment-threshold 160;
                }
                nc {
                    no-fragmentation;
                }
            }
        }
        schedulers {
            economy {
                transmit-rate percent 26;
                buffer-size percent 26;
            }
            business {
                transmit-rate percent 26;
                buffer-size percent 26;
            }
            stream {
                transmit-rate percent 35;
                buffer-size percent 35;
            }
            voice {
                transmit-rate percent 13;
                buffer-size percent 13;
            }
            link-economy {
                transmit-rate percent 26;
                buffer-size percent 26;
            }
            link-business {
                transmit-rate percent 26;
                buffer-size percent 26;
            }
            link-stream {
                transmit-rate percent 35;
                buffer-size percent 35;
            }
            link-voice {
                transmit-rate percent 13;
                buffer-size percent 13;
            }
        }
    }
}
```

SEE ALSO

- Layer 2 Service Package Capabilities and Interfaces | 921
- Link Services Configuration for Junos Interfaces | 847

Configuring LSQ Interfaces for T3 Links Configured for Compressed RTP over MLPPP

This example bundles a single T3 interface on a link services IQ interface with MLPPP encapsulation. Binding a single T3 interface to a multilink bundle allows you to configure compressed RTP (CRTP) on the T3 interface.

This scenario applies to MLPPP bundles only. The Junos OS does not currently support CRTP over Frame Relay. For more information, see "Configuring Services Interfaces for Voice Services" on page 1047 .

There is no need to configure LFI at DS3 speeds, because the packet serialization delay is negligible.

```
[edit interfaces]
t3-0/0/0 {
    unit 0 {
        family mlppp {
            bundle lsq-1/3/0.1;
        }
    }
}
lsq-1/3/0.1 {
    encapsulation multilink-ppp;
}
compression {
    rtp {
        # cRTP parameters go here
        #
        port minimum 2000 maximum 64009;
    }
}
```

This configuration uses a default fragmentation map, which results in all forwarding classes (queues) being sent out with a multilink header.

To eliminate multilink headers, you can configure a fragmentation map in which all queues have the `no-fragmentation` statement at the `[edit class-of-service fragmentation-maps` *`map-name`* `forwarding-class` *`class-name`*`]` hierarchy level, and attach the fragmentation map to the `lsq-1/3/0.1` interface, as shown here:

```
[edit class-of-service]
fragmentation-maps {
    fragmap {
        forwarding-class {
            be {
                no-fragmentation;
            }
            af {
                no-fragmentation;
            }
            ef {
                no-fragmentation;
            }
            nc {
                no-fragmentation;
            }
        }
    }
}
interfaces {
    lsq-1/3/0.1 {
        fragmentation-map fragmap;
    }
}
```

SEE ALSO

Layer 2 Service Package Capabilities and Interfaces | **921**

Link Services Configuration for Junos Interfaces | **847**

Configuring LSQ Interfaces as T3 or OC3 Bundles Using FRF.12

This example configures a clear-channel T3 or OC3 interface with multiple logical interfaces (DLCIs) on the link. In this scenario, each DLCI represents a customer. DLCIs are shaped at the egress PIC to a particular speed (*N*xDS0). This allows you to configure LFI using FRF.12 End-to-End Protocol on Frame Relay DLCIs.

To do this, first configure logical interfaces (DLCIs) on the physical interface. Then bundle the DLCIs, so that there is only one DLCI per bundle.

The physical interface must be capable of per-DLCI scheduling, which allows you to attach shaping rates to each DLCI. For more information, see the Junos OS Network Interfaces Library for Routing Devices.

To prevent fragment drops at the egress PIC, you must assign a shaping rate to the link services IQ logical interfaces and to the egress DLCIs. Shaping rates on DLCIs specify how much bandwidth is available for each DLCI. The shaping rate on link services IQ interfaces should match the shaping rate assigned to the DLCI that is associated with the bundle.

Egress interfaces also must have a scheduler map attached. The queue that carries voice should be strict-high-priority, while all other queues should be low-priority. This makes LFI possible.

This example shows voice traffic in the `ef` queue. The voice traffic is interleaved with bulk data. Alternatively, you can use multiclass MLPPP to carry multiple classes of traffic in different multilink classes.

```
[edit interfaces]
t3-0/0/0 {
    per-unit-scheduler;
    encapsulation frame-relay;
    unit 0 {
        dlci 69;
        family mlfr-end-to-end {
            bundle lsq-1/3/0.0;
        }
    }
    unit 1 {
        dlci 42;
        family mlfr-end-to-end {
            bundle lsq-1/3/0.1;
        }
    }
}
lsq-1/3/0 {
    unit 0 {
        encapsulation multilink-frame-relay-end-to-end;
    }
    fragment-threshold 320; # Multilink packets must be fragmented
}
unit 1 {
    encapsulation multilink-frame-relay-end-to-end;
}
```

```
fragment-threshold 160;
[edit class-of-service]
scheduler-maps {
    sched { # Scheduling parameters that apply to bundles on AS or Multiservices PICs.
    ...
}
pic-sched {
    # Scheduling parameters for egress DLCIs.
    # The voice queue should be strict-high priority.
    # All other queues should be low priority.
    ...
}
fragmentation-maps {
    fragmap {
        forwarding-class {
            ef {
                no-fragmentation;
            }
            # Voice is carried in the ef queue.
            # It is interleaved with bulk data.
        }
    }
}
interfaces {
    t3-0/0/0 {
        unit 0 {
            shaping-rate 512k;
            scheduler-map pic-sched;
        }
        unit 1 {
            shaping-rate 128k;
            scheduler-map pic-sched;
        }
    }
    lsq-1/3/0 { # Assign fragmentation and scheduling to LSQ interfaces.
    unit 0 {
        shaping-rate 512k;
        scheduler-map sched;
        fragmentation-map fragmap;
    }
    unit 1 {
        shaping-rate 128k;
        scheduler-map sched;
        fragmentation-map fragmap;
    }
}
```

For more information about how FRF.12 works with links services IQ interfaces, see "Configuring LSQ Interfaces for Single Fractional T1 or E1 Interfaces Using FRF.12" on page 939 .

SEE ALSO

Layer 2 Service Package Capabilities and Interfaces | 921

Link Services Configuration for Junos Interfaces | 847

Configuring LSQ Interfaces for ATM2 IQ Interfaces Using MLPPP

This example configures an ATM2 IQ interface with MLPPP bundled with link services IQ interfaces. This allows you to configure LFI on ATM virtual circuits.

For this type of configuration, the ATM2 IQ interface must have LLC encapsulation.

The following ATM PICs are supported in this scenario:

- 2-port OC-3/STM1 ATM2 IQ

- 4-port DS3 ATM2 IQ

Virtual circuit multiplexed PPP over AAL5 is not supported. Frame Relay is not supported. Bundling of multiple ATM VCs into a single logical interface is not supported.

Unlike DS3 and OC3 interfaces, there is no need to create a separate scheduler map for the ATM PIC. For ATM, you define CoS components at the `[edit interfaces at-fpc/pic/port atm-options]` hierarchy level, as described in the Junos OS Network Interfaces Library for Routing Devices.

NOTE: Do not configure RED profiles on ATM logical interfaces that are bundled. Drops do not occur at the ATM interface.

In this example, two ATM VCs are configured and bundled into two link services IQ bundles. A fragmentation map is used to interleave voice traffic with other multilink traffic. Because MLPPP is used, each link services IQ bundle can be configured for CRTP.

```
[edit interfaces]
at-1/2/0 {
    atm-options {
        vpi 0;
        pic-type atm2;
    }
    unit 0 {
        vci 0.69;
        encapsulation atm-mlppp-llc;
        family mlppp {
            bundle lsq-1/3/0.10;
        }
    }
    unit 1 {
        vci 0.42;
        encapsulation atm-mlppp-llc;
        family mlppp {
            bundle lsq-1/3/0.11;
        }
    }
}
lsq-1/3/0 {
    unit 10 {
        encapsulation multilink-ppp;
    }
    # Large packets must be fragmented.
    # You can specify fragmentation for each forwarding class.
    fragment-threshold 320;
    compression {
        rtp {
            port minimum 2000 maximum 64009;
        }
    }
}
unit 11 {
    encapsulation multilink-ppp;
}
fragment-threshold 160;
[edit class-of-service]
scheduler-maps {
    sched { # Scheduling parameters that apply to LSQ bundles on AS or Multiservices PICs.
    ...
}
fragmentation-maps {
    fragmap {
        forwarding-class {
            ef {
                no-fragmentation;
            }
        }
    }
}
interfaces { # Assign fragmentation and scheduling parameters to LSQ interfaces.
lsq-1/3/0 {
    unit 0 {
        shaping-rate 512k;
        scheduler-map sched;
        fragmentation-map fragmap;
    }
    unit 1 {
        shaping-rate 128k;
        scheduler-map sched;
        fragmentation-map fragmap;
    }
}
```

SEE ALSO

- Layer 2 Service Package Capabilities and Interfaces | **921**
- Link Services Configuration for Junos Interfaces | **847**

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
15.1	Starting in Junos OS Release 15.1, you can configure inline MLPPP interfaces on MX80, MX104, MX240, MX480, and MX960 routers with Channelized E1/T1 Circuit Emulation MICs.

11

PART

Traffic Load Balancer

Traffic Load Balancer | 984

CHAPTER 38

Traffic Load Balancer

IN THIS CHAPTER

- Traffic Load Balancer | 984

Traffic Load Balancer

IN THIS SECTION

- Traffic Load Balancer Overview | 984
- Configuring TLB | 993

Traffic Load Balancer Overview

IN THIS SECTION

- Traffic Load Balancing Support Summary | 985
- Traffic Load Balancer Application Description | 986
- Traffic Load Balancer Modes of Operation | 987
- Traffic Load Balancer Functions | 989
- Traffic Load Balancer Application Components | 990
- Traffic Load Balancer Configuration Limits | 992

Traffic Load Balancing Support Summary

Table 26 on page 985 provides a summary of the traffic load balancing support on the MS-MPC and MS-MIC cards for Adaptive Services versus support on the MX-SPC3 security services card for Next Gen Services.

Table 26: Traffic Load Balancing Support Summary

	MS-MPC		MX-SPC3
Junos Release	< 16.1R6 & 18.2.R1	≥ 16.1R6 & 18.2R1	19.3R2
Max # of Instances per Chassis	32	2,000 / 32 in L2 DSR mode	2,000
Max # of Virtual Services per Instance	32	32	32
Max # of virtual IP address per virtual service		1	1
Max # of Groups per Instances	32	32	32
Max # of Real-Services (Servers) per Group	255	255	255
Max # of groups per virtual service		1	1
Max # of Network Monitor Profiles per Group		2	2
Max # of HC’s per security services per PIC/NPU in 5-sec’s		4,000	1,250 – 19.3R2 10,000 – 20.1R1
Supported Health Check Protocols	ICMP, TCP, UDP, HTTP, SSL, Custom		ICMP, TCP, UDP, HTTP, SSL, TLS Hello, Custom

Traffic Load Balancer Application Description

Traffic Load Balancer (TLB) is supported on MX Series routers with either of the Multiservices Modular Port Concentrator (MS-MPC), Multiservices Modular Interface Card (MS-MIC), or the MX Security Services Processing Card (MX-SPC3) and in conjunction with the Modular Port Concentrator (MPC) line cards supported on the MX Series routers as described in Table 27 on page 986 .

> **NOTE**: You cannot run Deterministic NAT and TLB simultaneously.

Table 27: TLB MX Series Router Platform Support Summary

TLB Mode	MX Platform Coverage
Multiservices Modular Port Concentrator (MS-MPC)	MX240, MX2480, MX960, MX2008, MX2010, MX2020
MX Security Services Processing Card (MX-SPC3)	MX240, MX480, MX960

- TLB enables you to distribute traffic among multiple servers.
- TLB employs an MS-MPC-based control plane and a data plane using the MX Series router forwarding engine.
- TLB uses an enhanced version of equal-cost multipath (ECMP). Enhanced ECMP facilitates the distribution of flows across groups of servers. Enhancements to native ECMP ensure that when servers fail, only flows associated with those servers are impacted, minimizing the overall network churn on services and sessions.
- TLB provides application-based health monitoring for up to 255 servers per group, providing Intelligent traffic steering based on health checking of server availability information. You can configure an aggregated multiservices (AMS) interface to provide one-to-one redundancy for MS-MPCs or Next Gen Services MX-SPC3 card used for server health monitoring.
- TLB applies its flow distribution processing to ingress traffic.
- TLB supports multiple virtual routing instances to provide improved support for large scale load balancing requirements.
- TLB supports static virtual-IP-address-to-real-IP-address translation, and static destination port translation during load balancing.

Traffic Load Balancer Modes of Operation

Traffic Load Balancer provides three modes of operation for the distribution of outgoing traffic and for handling the processing of return traffic.

Table 28 on page 987 summarizes the TLB support and which cards it's supported on.

Table 28: TLB Versus Security Service Cards Summary

Security Service Card	MS-MPC	MX-SPC3
Translate	Yes	Yes
Transparent Layer 3 Direct Server Return	Yes	Yes
Transparent Layer 2 Direct Server Return	Yes	Not Supported

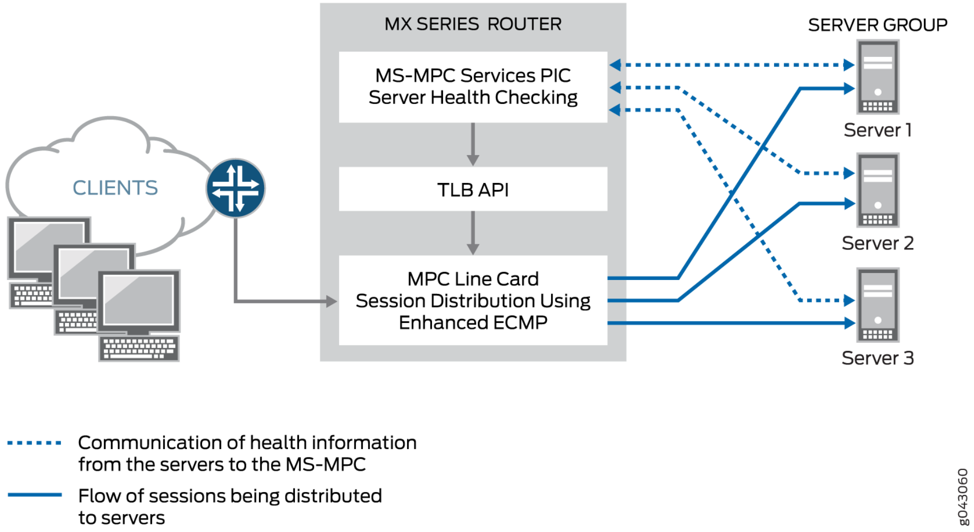
Transparent Mode Layer 2 Direct Server Return

When you use transparent mode Layer 2 direct server return (DSR):

- The PFE processes data.
- Load balancing works by changing the Layer 2 MAC of packets.
- An MS-MPC performs the network-monitoring probes.
- Real servers must be directly (Layer 2) reachable from the MX Series router.
- TLB installs a route and all the traffic over that route is load-balanced.
- TLB never modifies Layer 3 and higher level headers.

Figure 46 on page 988 shows the TLB topology for transparent mode Layer 2 DSR.

Figure 46: TLB Topology for Transparent Mode



Translated Mode

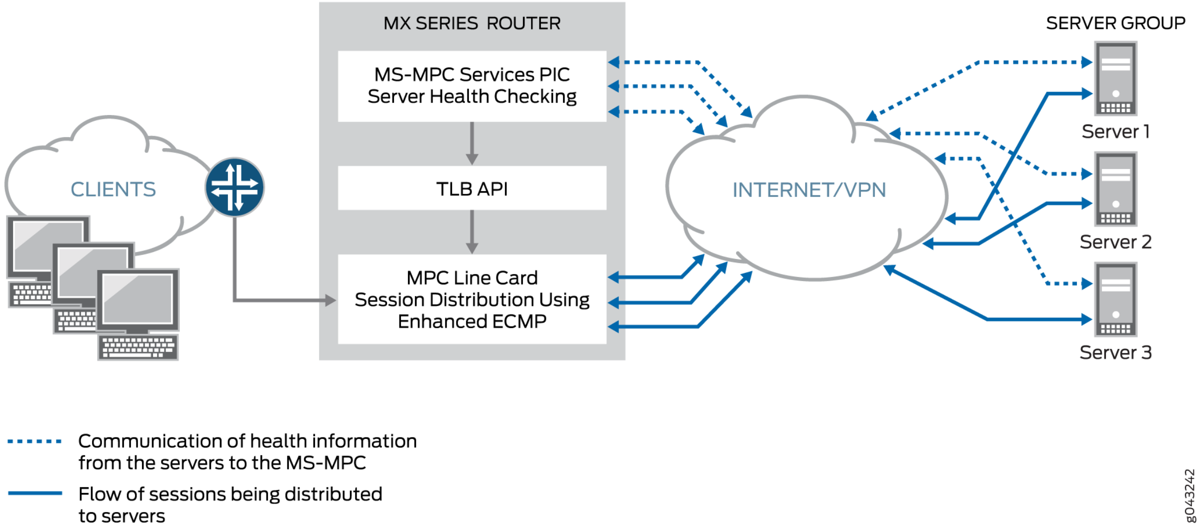
Translated mode provides greater flexibility than transparent mode Layer 2 DSR. When you choose translated mode:

- An MS-MPC performs the network-monitoring probes.
- The PFE performs stateless load balancing:
  - Data traffic directed to a virtual IP address undergoes translation of the virtual IP address to a real server IP address and translates the virtual port to a server listening port. Return traffic undergoes the reverse translation.
  - Client to virtual IP traffic is translated; the traffic is routed to reach its destination.
  - Server-to-client traffic is captured using implicit filters and directed to an appropriate load-balancing next hop for reverse processing. After translation, traffic is routed back to the client.
  - Two load balancing methods are available: random and hash. The random method is only for UDP traffic and provides quavms-random distribution. While not literally random, this mode provides fair distribution of traffic to an available set of servers. The hash method provides a hash key based on any combination of the source IP address, destination IP address, and protocol.

> **NOTE**: Translated mode processing is only available for IPv4-to-IPv4 and IPv6-to-IPv6 traffic.

Figure 47 on page 989 shows the TLB topology for translated mode.

Figure 47: TLB Topology for Translated Mode



Transparent Mode Layer 3 Direct Server Return

Transparent mode Layer 3 DSR load balancing distributes sessions to servers that can be a Layer 3 hop away. Traffic is returned directly to the client from the real-server.

Traffic Load Balancer Functions

TLB provides the following functions:

- TLB always distributes the *requests* for any flow. When you specify DSR mode, the response returns directly to the source. When you specify translated mode, reverse traffic is steered through implicit filters on server-facing interfaces.
- TLB supports hash-based load balancing or random load balancing.
- TLB enables you to configure servers offline to prevent a performance impact that might be caused by a rehashing for all existing flows. You can add a server in the administrative down state and use it later for traffic distribution by disabling the administrative down state. Configuring servers offline helps prevent traffic impact to other servers.
- When health checking determines a server to be down, only the affected flows are rehashed.
- When a previously down server is returned to service, all flows belonging to that server based on hashing return to it, impacting performance for the returned flows. For this reason, you can disable

the automatic rejoining of a server to an active group. You can return servers to service by issuing the `request services traffic-load-balance real-service rejoin` operational command.

> **NOTE**: NAT is not applied to the distributed flows.

- Health check monitoring application runs on an MS-MPC/NPU. This network processor unit (NPU) is not used for handling data traffic.
- TLB supports static virtual-IP-adddress-to-real-IP-address translation, and static destination port translation during load balancing.
- TLB provides multiple VRF support.

Traffic Load Balancer Application Components

Servers and Server Groups

TLB enables configuration of groups of up to 255 servers (referred to in configuration statements as *real services*) for use as alternate destinations for stateless session distribution. All servers used in server groups must be individually configured before assignment to groups. Load balancing uses hashing or randomization for session distribution. Users can add and delete servers to and from the TLB server distribution table and can also change the administrative status of a server.

> **NOTE**: TLB uses the session distribution next-hop API to update the server distribution table and retrieve statistics. *Applications do not have direct control on the server distribution table management. They can only influence changes indirectly through the add and delete services of the TLB API.*

Server Health Monitoring — Single Health Check and Dual Health Check

TLB supports TCP, HTTP, SSL Hello, TLS Hello, and custom health check probes to monitor the health of servers in a group. You can use a single probe type for a server group, or a dual health check configuration that includes two probe types. The configurable health monitoring function resides on either an MX-SPC3 or an MS-MPC. By default, probe requests are sent every 5 seconds. Also by default, a real server is declared down only after five consecutive probe failures and declared up only after five consecutive probe successes.

Use a custom health check probe to specify the following:

- Expected string in the probe response

- String that is sent with the probe
- Server status to assign when the probe times out (up or down)
- Server status to assign when the expected response to the probe is received (up or down)
- Protocol — UDP or TCP

TLB provides *application stickiness*, meaning that server failures or changes do not affect traffic flows to other active servers. Changing a server's administrative state from up to down does not impact any active flows to remaining servers in the server distribution table. Adding a server or deleting a server from a group has some traffic impact for a length of time that depends on your configuration of the interval and retry parameters in the monitoring profile.

TLB provides two levels of server health monitoring:

- Single Health Check—One probe type is attached to a server group by means of the `network-monitoring-profile` *configuration statement*.
- TLB Dual Health Check (TLB-DHC)—Two probe types are associated with a server group by means of the `network-monitoring-profile` configuration statement. A server's status is declared based on the result of two health check probes. Users can configure up to two health check profiles per server group. If a server group is configured for dual health check, a real-service is declared to be UP only when both health-check probes are simultaneously UP; otherwise, a real-service is declared to be DOWN.

NOTE: The following restrictions apply to AMS interfaces used for server health monitoring:

- An AMS interface configured under a TLB instance uses its configured member interfaces exclusively for health checking of configured multiple real servers.
- The member interfaces use unit 0 for single VRF cases, but can use units other than 1 for multiple VRF cases.
- TLB uses the IP address that is configured for AMS member interfaces as the source IP address for health checks.
- The member interfaces must be in the same routing instance as the interface used to reach real servers. This is mandatory for TLB server health-check procedures.

Virtual Services

The virtual service provides a virtual IP address (VIP) that is associated with the group of servers to which traffic is directed as determined by hash-based or random session distribution and server health monitoring. In the case of Layer2 DSR and Layer3 DSR, the special address 0.0.0.0 causes all traffic flowing to the forwarding instance to be load balanced.

The virtual service configuration includes:

- Mode—indicating how traffic is handled (translated or transparent).
- The group of servers to which sessions are distributed.
- The load balancing method.
- Routing instance and route metric.

BEST PRACTICE: Although you can assign a virtual address of 0.0.0.0 in order to use default routing, we recommend using a virtual address that can be assigned to a routing instance set up specifically for TLB.

Traffic Load Balancer Configuration Limits

Traffic Load Balancer configuration limits are described in Table 29 on page 992 .

Table 29: TLB Configuration Limits

Configuration Component	Configuration Limit
Maximum number of instances.	Starting in Junos OS Release 16.1R6 and Junos OS Release 18.2R1, the TLB application supports 2000 TLB instances for virtual services that use the direct-server-return or the translated mode. In earlier releases, the maximum number of instances is 32. If multiple virtual services are using the same server group, then all of those virtual services must use the same load balancing method to support 2000 TLB instances. For virtual services that use the layer2-direct-server-return mode, TLB supports only 32 TLB instances. To perform the same function as the layer2-direct-server-return mode and have support for 2000 TLB instances, you can use the direct-server-return mode and use a service filter with the skip action.
Maximum number of servers per group	255

Table 29: TLB Configuration Limits *(Continued)*

Configuration Component	Configuration Limit
Maximum number of virtual services per services PIC	32
Maximum number of health checks per services PIC in a 5-second interval	For MS-MPC services cards: 2000 For Next Gen Services mode and the MX-SPC3 services cards: 1250
Maximum number of groups per virtual service	1
Maximum number of virtual IP addresses per virtual service	1
Supported health checking protocols	ICMP, TCP, HTTP, SSL, TLS-Hello, Custom **NOTE**: ICMP health checking is supported only on MS-MPC services cards. Starting in Junos OS release 22.4.1, TLB is enhanced to support TLS-Hello health check type. For TLS-Hello over TCP, TLS v1.2 and v1.3 TLS-Hello health checks are supported.

SEE ALSO

Interchassis High-Availability

Understanding AMS Interfaces

Configuring TLB

IN THIS SECTION

- Loading the TLB Service Package | 994
- Configuring a TLB Instance Name | 994
- Configuring Interface and Routing Information | 995

- Configuring Servers | 998
- Configuring Network Monitoring Profiles | 998
- Configuring Server Groups | 1000
- Configuring Virtual Services | 1002
- Configuring Tracing for the Health Check Monitoring Function | 1005

The following topics describe how to configure TLB. To create a complete application, you must also define interfaces and routing information. You can optionally define firewall filters and policy options in order to differentiate TLB traffic.

Loading the TLB Service Package

Load the TLB service package on each service PIC on which you want to run TLB.

> **NOTE**: For Next Gen Services and the MX-SPC3 services card, you do not need to load this package.

To load the TLB service package on a service PIC:

- Load the `jservices-traffic-dird` package.

  ```
  [edit chassis fpc slot-number pic pic-number adaptive-services service-package extension-provider]
  user@host# set package jservices-traffic-dird
  ```

  For example:

  ```
  [edit chassis fpc 3 pic 0 adaptive-services service-package extension-provider]
  user@host# set package jservices-traffic-dird
  ```

Configuring a TLB Instance Name

Before configuring TLB, enable the sdk-service process by configuring `system processes sdk-service enable` at the [edit] hierarchy.

To configure a name for the TLB instance:

- At the [edit services traffic-load-balance] hierarchy level, identify the TLB instance name.

  ```
  [edit services traffic-load-balance]
  user@host# set instance instance-name
  ```

  For example:

  ```
  [edit services traffic-load-balance]
  user@host# set instance tlb-instance1
  ```

Configuring Interface and Routing Information

To configure interface and routing information:

1. At the [edit services traffic-load-balance instance *instance-name*] hierarchy level, identify the service interface associated with this instance.

   ```
   user@host# [edit services traffic-load-balance instance instance-name]
   user@host# set interface interface-name
   ```

   For example, on an MS-MPC:

   ```
   [edit services traffic-load-balance instance tlb-instance1]
   user@host# set interface ms-1/0/0
   ```

   For example, for Next Gen Services on an MX-SPC3:

   ```
   [edit services traffic-load-balance instance tlb-instance1]
   user@host# set interface vms-1/0/0
   ```

2. Enable the routing of health-check packet responses from real servers to the service interface that you identified in Step 1.

   ```
   [edit interfaces]
   user@host# set interface-name unit 0 ip-address-owner service-plane
   ```

For example, on an MS-MPC:

```
[edit interfaces]
user@host# set ms-1/0/0 unit 0 ip-address-owner service-plane
```

For example, on an MX-SPC3:

```
[edit interfaces]
user@host# set vms-1/0/0 unit 0 ip-address-owner service-plane
```

3. Specify the client interface for which an implicit filter is defined to direct traffic in the forward direction. This is required only for translated mode.

```
user@host# [edit services traffic-load-balance instance instance-name]
user@host# set client-interface interface-name
```

For example:

```
[edit services traffic-load-balance instance tlb-instance1]
user@host# set client-interface ge-5/2/0.0
```

4. Specify the virtual routing instance used to route data traffic in the forward direction to servers. This is required for SLT and Layer 3 DSR; it is optional for Layer 2 DSR.

```
user@host# [edit services traffic-load-balance instance instance-name]
user@host# set server-vrf server-vrf
```

For example:

```
[edit services traffic-load-balance instance tlb-instance1]
user@host# set server-vrf server-vrf
```

5. Specify the server interface for which implicit filters are defined to direct return traffic to the client.

NOTE: Implicit filters for return traffic are not used for DSR.

```
user@host# [edit services traffic-load-balance instance instance-name]
user@host# set server-interface server-interface
```

For example:

```
[edit services traffic-load-balance instance tlb-instance1]
user@host# set server-interface ge-5/2/1.0
```

6. (Optional) Specify the filter used to bypass health checking for return traffic.

```
user@host# [edit services traffic-load-balance instance instance-name]
user@host# set server-inet-bypass-filter server-inet-bypass-filter
```

For example:

```
[edit services traffic-load-balance instance tlb-instance1]
user@host# set server-inet-bypass-filter tlb-ipv4-bypass
```

7. Specify the virtual routing instance in which you want the data in the reverse direction to be routed to the clients.

```
user@host# [edit services traffic-load-balance instance instance-name]
user@host# set client-vrf client-vrf
```

For example:

```
[edit services traffic-load-balance instance tlb-instance1]
user@host# set client-vrf client-vrf
```

NOTE: Virtual routing instances for routing data in the reverse direction are not used with DSR.

Configuring Servers

To configure servers for the TLB instance:

- Configure a logical name and IP address for each server to be made available for next-hop distribution.

  ```
  [edit services traffic-load-balance instance instance-name]
  user@host# set real-service real-service-name address server-ip-address
  ```

  For example:

  ```
  [edit services traffic-load-balance instance tlb-instance1]
  user@host# set real-service rs138 address 172.26.99.138
  user@host# set real-service rs139 address 172.26.99.139
  user@host# set real-service rs140 address 172.26.99.140
  ```

Configuring Network Monitoring Profiles

A network monitoring profile configures a health check probe, which you assign to a server group to which session traffic is distributed.

To configure a network monitoring profile:

1. Configure the type of probe to use for health monitoring — `icmp`, `tcp`, `http`, `ssl-hello`, `tls-hello`,or `custom`.

   NOTE: `icmp` probes are supported only on MS-MPC cards.
   Next Gen Services and the MX-SPC3 do not support ICMP probes in this release.

   - For an ICMP probe:

     ```
     [edit services network-monitoring profile profile-name]
     user@host.com# set icmp
     ```

   - For a TCP probe:

     ```
     [edit services network-monitoring profile profile-name]
     user@host.com# set tcp port tcp-port-number
     ```

- For an HTTP probe:

  ```
  [edit services network-monitoring profile profile-name]
  user@host.com# set http host hostname url url port http-port-number method (get | option)
  ```

- For an SSL probe:

  ```
  [edit services network-monitoring profile profile-name]
  user@host.com# set ssl-hello port port ssl-version
  ```

- For a TLS-Hello probe:

  ```
  [edit services network-monitoring profile profile-name]
  user@host.com# set tls-hello port port number
  ```

- For a custom probe:

  ```
  [edit services network-monitoring profile profile-name]
  user@host.com# set custom cmd priority default-real-service-status (down | up) expect
  (ascii | binary) receive-string port port real-service-action (down | up) send (ascii |
  binary) send-string
  ```

2. Configure the interval for probe attempts, in seconds (1 through 180).

   ```
   [edit services network-monitoring profile profile-name]
   user@host.com# set probe-interval interval
   ```

   For example:

   ```
   [edit services network-monitoring profile profile1-icmp]
   user@host.com# set probe-interval 2
   ```

3. Configure the number of failure retries, after which the real server is tagged as down.

   ```
   [edit services network-monitoring profile profile-name]
   user@host.com# set failure-retries number-of-retries
   ```

For example:

```
[edit services network-monitoring profile profile1-icmp]
user@host.com# set failure-retries 3
```

4. Configure the number of recovery retries, which is the number of successful probe attempts after which the server is declared up.

```
[edit services network-monitoring profile profile-name]
user@host.com# set recovery-retries number-of-retries
```

For example:

```
[edit services network-monitoring profile profile1-icmp]
user@host.com# set recovery-retries 1
```

Configuring Server Groups

Server groups consist of servers to which traffic is distributed by means of stateless, hash-based session distribution and server health monitoring.

To configure a server group:

1. Specify the names of one or more configured real servers.

```
[edit services traffic-load-balance instance instance-name groups group-name]
user@host.com# set real-services real-service-name, ...
```

For example:

```
[edit services traffic-load-balance instance tlb-instance1 groups tlb-group1]
user@host.com# set real-services [ rs138 rs139 rs140 ]
```

2. Configure the routing instance for the group when you do not want to use the default instance, `inet.0`.

```
[edit services traffic-load-balance instance instance-name groups group-name]
user@host.com# set routing-instance routing-instance-name
```

For example:

```
[edit services traffic-load-balance instance tlb-instance1 groups tlb-group1]
user@host.com# set routing-instance tlb-routing-instance1
```

3. (Optional) Disable the default option that allows a server to rejoin the group automatically when it comes up.

```
[edit services traffic-load-balance instance instance-name group group-name]
user@host.com# set real-service-rejoin-options no-auto-rejoin
```

4. (Optional) Configure the logical unit of the instance's service interface to use for health checking.

   a. Specify the logical unit.

   ```
   [edit services traffic-load-balance instance instance-name group group-name]
   user@host.com# set health-check-interface-subunit health-check-interface-subunit
   ```

   b. Enable the routing of health-check packet responses from real servers to the interface.

   ```
   [edit interfaces]
   user@host.com# set interface-name unit subunit ip-address-owner service-plane
   ```

For example:

```
[edit services traffic-load-balance instance tlb-instance1 group tlb-group1]
user@host.com# set health-check-interface-subunit 30
[edit interfaces]
user@host.com# set ms-1/0/0 unit 30 ip-address-owner service-plane
```

5. Configure one or two network monitoring profiles to be used to monitor the health of servers in this group.

```
[edit services traffic-load-balance instance instance-name groups group-name]
user@host.com# set network-monitoring-profile profile-name1 profile-name2
```

For example:

```
[edit services traffic-load-balance instance tlb-instance1 groups tlb-group1]
user@host.com# set network-monitoring-profile profile1-icmp profile2-http
```

Configuring Virtual Services

A virtual service provides an address that is associated with a the group of servers to which traffic is directed as determined by hash-based or random session distribution and server health monitoring. You may optionally specify filters and routing instances to steer traffic for TLB.

To configure a virtual service:

1. At the `[edit services traffic-load-balance instance instance-name]` hierarchy level, specify a non-zero address for the virtual service.

   ```
   [edit services traffic-load-balance instance instance-name virtual-service virtual-service-
   name]
   user@host# set address virtual-ip-address
   ```

   For example:

   ```
   [edit services traffic-load-balance instance tlb-instance1 virtual-service virtual-service1]
   user@host# set address 192.0.2.11
   ```

2. Specify the server group used for this virtual service.

   ```
   [edit services traffic-load-balance instance instance-name virtual-service virtual-service-
   name]
   user@host# set group group-name
   ```

   For example:

   ```
   [edit services traffic-load-balance instance tlb-instance1  virtual-service virtual-service1]
   user@host# set group tlb-group1
   ```

3. (Optional) Specify a routing instance for the virtual service. If you do not specify a routing instance, the default routing instance is used.

   ```
   [edit services traffic-load-balance instance instance-name virtual-service virtual-service-name]
   user@host# set routing-instance routing-instance
   ```

   For example:

   ```
   [edit services traffic-load-balance instance tlb-instance1  virtual-service virtual-service1]
   user@host# set routing-instance msp-tproxy-server-vrf31
   ```

4. Specify the processing mode for the virtual service.

   ```
   [edit services traffic-load-balance instance instance-name virtual-service virtual-service-name]
   user@host# set mode (layer2-direct-server-return | direct-server-return | translated)
   ```

   For example:

   ```
   [edit services traffic-load-balance instance tlb-instance1  virtual-service virtual-service1]
   user@host# set mode translated
   ```

5. (Optional) For a translated mode virtual service, enable the addition of the IP addresses for all the real servers in the group under the virtual service to the server-side filters. Doing this allows you to configure two virtual services with the same listening port and protocol on the same interface and VRF.

   ```
   [edit services traffic-load-balance instance instance-name virtual-service virtual-service-name]
   user@host# set include-real-server-ips-in-server-filter
   ```

6. (Optional) Specify a routing metric for the virtual service.

   ```
   [edit services traffic-load-balance instance instance-name virtual-service virtual-service-name]
   user@host# set routing-metric routing-metric
   ```

For example:

```
[edit services traffic-load-balance instance tlb-instance1  virtual-service virtual-service1]
user@host# set routing-metric 128
```

7. Specify the method used for load balancing. You can specify a hash method that provides a hash key based on any combination of the source IP address, destination IP address, and protocol, or you can specify `random`.

```
[edit services traffic-load-balance instance tlb-instance1  virtual-service virtual-service1]
user@host# set load-balancing-method (hash hash-key (source-ip | destination-ip | proto) |
random)
```

For example:

```
[edit services traffic-load-balance instance tlb-instance1  virtual-service virtual-service1]
user@host# set load-balancing-method hash hash-key source-ip
```

or

```
[edit services traffic-load-balance instance tlb-instance1  virtual-service virtual-service1]
user@host# set load-balancing-method random
```

NOTE: If you switch between the hash method and the random method for a virtual service, the statistics for the virtual service are lost.

8. For a translated mode virtual service, specify a service for translation, including a virtual-port, server-listening-port, and protocol.

```
[edit services traffic-load-balance instance instance-name virtual-service virtual-service-
name]
user@host# set service service-name virtual-port virtual-port server-listening-port server-
listening-port protocol (udp | tcp)
```

For example:

```
[edit services traffic-load-balance instance tlb-instance1  virtual-service virtual-service1]
user@host# set service fast-track-service virtual-port 1111 server-listening-port 22 protocol
tcp
```

9. Commit the configuration.

```
[edit services traffic-load-balance instance instance-name virtual-service virtual-service-
name]
user@host# commit
```

NOTE: In the absence of a client-interface configuration under the TLB instance, the implicit client filter (for VIP) is attached to the client-vrf configured under the TLB instance. In this case, the routing-instance under a translate mode virtual service cannot be the same as the client-vrf configured under the TLB instance. if it is, the commit fails.

Configuring Tracing for the Health Check Monitoring Function

To configure tracing options for the health check monitoring function:

1. Specify that you want to configure tracing options for the health check monitoring function.

```
[edit services network-monitoring]
user@host# edit traceoptions
```

2. (Optional) Configure the name of the file used for the trace output.

```
[edit services  network-monitoring traceoptions]
user@host# set file file-name
```

3. (Optional) Disable remote tracing capabilities.

```
[edit services  network-monitoring traceoptions]
user@host# set no-remote-trace
```

4. (Optional) Configure flags to filter the operations to be logged.

```
[edit services  network-monitoring traceoptions]
user@host# set flag flag
```

Table 30 on page 1006 describes the flags that you can include.

Table 30: Trace Flags

Flag	Support on MS-MPC and MX-SPC3 Cards	Description
all	MS-MPC and MX-SPC3	Trace all operations.
all-real-services	MX-SPC3	Trace all real services.
config	MS-MPC and MX-SPC3	Trace traffic load balancer configuration events.
connect	MS-MPC and MX-SPC3	Trace traffic load balancer ipc events.
database	MS-MPC and MX-SPC3	Trace database events.
file-descriptor-queue	MS-MPC	Trace file descriptor queue events.
inter-thread	MS-MPC	Trace inter-thread communication events.
filter	MS-MPC and MX-SPC3	Trace traffic load balancer filter programming events.
health	MS-MPC and MX-SPC3	Trace traffic load balancer health events.
messages	MS-MPC and MX-SPC3	Trace normal events.
normal	MS-MPC and MX-SPC3	Trace normal events.
operational-commands	MS-MPC and MX-SPC3	Trace traffic load balancer show events.

Table 30: Trace Flags *(Continued)*

Flag	Support on MS-MPC and MX-SPC3 Cards	Description
parse	MS-MPC and MX-SPC3	Trace traffic load balancer parse events.
probe	MS-MPC and MX-SPC3	Trace probe events.
probe-infra	MS-MPC and MX-SPC3	Trace probe infra events.
route	MS-MPC and MX-SPC3	Trace traffic load balancer route events.
snmp	MS-MPC and MX-SPC3	Trace traffic load balancer SNMP events.
statistics	MS-MPC and MX-SPC3	Trace traffic load balancer statistics events.
system	MS-MPC and MX-SPC3	Trace traffic load balancer system events.

5. (Optional) Configure the level of tracing.

```
[edit services  network-monitoring traceoptions]
user@host# set level (all |error | info | notice | verbose | warning)
```

6. (Optional) Configure tracing for a particular real server within a particular server group.

```
[edit services network-monitoring traceoptions]
user@host# set monitor monitor-object-name group-name group-name real-services-name real-
service-name
```

7. (Optional) Starting in Junos OS Release 16.1R6 and 18.2R1, configure tracing for a particular virtual service and instance.

```
[edit services traffic-load-balance traceoptions]
user@host# set monitor monitor-object-name instance-name instance-name virtual-svc-name
virtual-service-name
```

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
16.1R6	Starting in Junos OS Release 16.1R6 and Junos OS Release 18.2R1, the TLB application supports 2000 TLB instances for virtual services that use the direct-server-return or the translated mode.
16.1R6	Starting in Junos OS Release 16.1R6 and 18.2R1, configure tracing for a particular virtual service and instance.

12

PART

Services Card Redundancy

Services Card Redundancy for MS-MPC and MS-MIC | 1010

Services Card Redundancy for Multiservices PIC | 1042

CHAPTER 39

Services Card Redundancy for MS-MPC and MS-MIC

IN THIS CHAPTER

- Load Balancing and High Availability With Aggregated Multiservices Interfaces on MS-MPC and MS-MIC | **1010**

Load Balancing and High Availability With Aggregated Multiservices Interfaces on MS-MPC and MS-MIC

IN THIS SECTION

- Understanding Aggregated Multiservices Interfaces | **1010**
- Configuring Aggregated Multiservices Interfaces | **1017**
- Configuring Load Balancing on AMS Infrastructure | **1020**
- Configuring Warm Standby for Services Interfaces | **1024**
- Example: Configuring an Aggregated Multiservices Interface (AMS) | **1025**
- Example: Configuring Next-Hop Style Services on an Aggregated Multiservices Interface | **1032**
- Example: Configuring Static Source Translation on AMS Infrastructure | **1037**

Understanding Aggregated Multiservices Interfaces

IN THIS SECTION

- Aggregated Multiservices Interface | **1011**

- IPv6 Traffic on AMS Interfaces Overview | **1014**
- Member Failure Options and High Availability Settings | **1016**
- Warm Standby Redundancy | **1017**

This topic contains the following sections:

Aggregated Multiservices Interface

In Junos OS, you can combine multiple services interfaces to create a bundle of services interfaces that can function as a single interface. Such a bundle of interfaces is known as an *aggregated multiservices interface* (AMS), and is denoted as ams*N* in the configuration, where *N* is a unique number that identifies an AMS interface (for example, ams0).

AMS configuration provides higher scalability, improved performance, and better failover and load-balancing options.

AMS configuration enables service sets to support multiple services PICs by associating an AMS bundle with a service set. An AMS bundle can have up to 24 services PICs as member interfaces and can distribute services among the member interfaces.

Member interfaces are identified as mams in the configuration. The chassisd process in routers that support AMS configuration creates a mams entry for every multiservices interface on the router.

Starting with Junos OS Release 16.2 (except Junos OS Release 17.3R3-S7), an AMS interface can have up to 36 member interfaces. If you include more than 24 member interfaces, you must increase the service PIC boot timeout to 240 or 300 seconds for all service PICs. In Junos OS Release 16.1 and earlier and in Junos OS Release 17.3R3-S7, an AMS interface could have a maximum of 24 member interfaces.

Starting with Junos OS Release 17.1R1, AMS supports IPSec tunnel distribution for next-hop style service-sets. However, interface-style IPSec service sets are not supported.

Starting with Junos OS Release 19.2R1, you can use up to 60 PICs across different AMS bundles on a MX2020 router. The hard limit of maximum 36 member interfaces per AMS bundle still exists. However, in the chassis, there can be multiple AMS bundles such that 15 MS-MPCs can be configured across these bundles.

When you configure services options at the ams interface level, the options apply to all member interfaces (mams) for the ams interface.

The options also apply to service sets configured on services interfaces corresponding to the ams interface's member interfaces. All settings are per PIC. For example, session-limit applies per member and not at an aggregate level.

Starting in Junos OS Release 19.3R2, AMS interfaces are supported with the MX-SPC3. The following table indicates the details of maximum number of MX-SPC3s, maximum number of PICs, and the maximum number of AMS members in a bundle:

MX Platforms	Maximum number of MX-SPC3s	Maximum number of PICs	Maximum number of AMS members
MX240	2	4	4
MX480	5	10	10
MX960	7	14	14

NOTE: You cannot configure services options at both the ams (aggregate) and member-interface level. If services options are configured on ms-x/y/z or vms-x/y/z, they also apply to service sets on mams-x/y/z.

When you want services options settings to apply uniformly to all members, configure services options at the ams interface level. If you need different settings for individual members, configure services options at the member interface level.

NOTE: Per-member drop of traffic and per-member next-hop configuration is required for NAT64. For NAPT44, this per-member specification allows arbitrary hash keys, providing better load-balancing options to allow dynamic NAT operations to be performed. For NAT64, NAPT44, and dynamic NAT44, it is not possible to determine which member allocates the dynamic NAT address. To ensure that reverse flow packets arrive at the same member as the forward flow packets, pool-address-based routes are used to steer reverse flow packets.

NOTE: Until Junos OS Release 13.3, for every media logical interface on which services were configured (interface style services), a logical interface alias was internally created. This interface alias stores the topology chains for features that are performed on the logical interface after an input service was processed to avoid packet loops in the system. With interface aliases, the maximum number of logical interfaces supported with services was reduced to half the supported maximum number because each logical interface consumed two entries, namely, one for the interface itself and the other for the interface alias.

Starting in Junos OS Release 14.1R4, input interface aliases are not created for MS-MPCs and MS-MICs. As a result, the maximum number of logical interfaces that are supported with services PICs is equal to the maximum number supported on the system. After input service processing by MS-MPCs and MS-MICs, the services PIC sends the packet to the Packet Forwarding Engine on the multiservices (`ms-`) logical interface where the corresponding service is configured. Post-services are not supported on MS- MPCs and MS-MICs in Junos OS Release 13.2 and later.

NOTE: You cannot include MS-DPCs or other MS-PICs in an AMS configuration that contains MS-MICs or MS-MPCs as member interfaces.

NOTE: If you modify a NAT pool that is being used by a service set assigned to an AMS interface, you must deactivate and activate the service set before the NAT pool changes take effect.

By default, the traffic distribution over the member interfaces of an AMS interface happens in a round-robin fashion. You can also configure the following hash key values to regulate the traffic distribution: `source-ip`, `destination-ip` , and `protocol`. For services that require traffic symmetry, you must configure symmetrical hashing. Symmetrical hashing configuration ensures that both forward and reverse traffic is routed through the same member interface.

With basic NAT44, load balancing on AMS interfaces of MS-MICs and MS-MPCs does not work properly if the ingress hash key is source IP address and the egress hash key is destination IP address.

If the service set is applied on the Gigabit Ethernet or 10-Gigabit Ethernet interface that functions as the NAT inside interface, then the hash keys used for load balancing might be configured in such a way that the ingress key is set as destination IP address and the egress key is set as source IP address. Because the source IP address undergoes NAT processing, it is not available for hashing the traffic in the reverse direction. Therefore, load balancing does not happen on the same IP address and forward and reverse traffic does not map to the same PIC. With the hash keys reversed, load balancing occurs correctly.

With next-hop services, for forward traffic, the ingress key on the inside interface load -balances traffic, and for reverse traffic, the ingress key on the outside interface load -balances traffic or per-member next hops steer reverse traffic. With interface-style services, the ingress key load-balances forward traffic and the egress key load-balances forward traffic or per-member next hops steer reverse traffic. Forward traffic is traffic entering from the inner side of a service set and reverse traffic is traffic entering from the outer side of a service set. The forward key is the hash key used for the forward direction of traffic and the reverse key is the hash key used for the reverse direction of traffic (depends on whether it relates to interface services or next-hop services style.)

With stateful firewalls, you can configure the following combinations of forward and reverse keys for load balancing. In the following combinations presented for hash keys, FOR-KEY refers to the forward key, REV-KEY denotes the reverse key, SIP signifies source IP address, DIP signifies destination IP address, and PROTO refers to protocol such as IP.

- FOR-KEY: SIP, REV-KEY: DIP
- FOR-KEY: SIP,PROTO REV-KEY: DIP, PROTO
- FOR-KEY: DIP, REV-KEY: SIP
- FOR-KEY: DIP,PROTO REV-KEY: SIP, PROTO
- FOR-KEY: SIP,DIP REV-KEY: SIP, DIP
- FOR-KEY: SIP,DIP,PROTO REV-KEY: SIP, DIP,PROTO

With static NAT configured as basic NAT44 or destination NAT44, and with stateful firewall configured or not, if the forward direction of traffic must undergo NAT processing, configure the hash keys as follows:

- FOR-KEY: DIP, REV-KEY: SIP
- FOR-KEY: DIP,PROTO REV-KEY: SIP, PROTO

If the reverse direction of traffic must undergo NAT processing, configure the hash keys as follows:

- FOR-KEY: SIP, REV-KEY: DIP
- FOR-KEY: SIP,PROTO REV-KEY: DIP, PROTO

With dynamic NAT configured, and with stateful firewall configured or not, only the forward direction traffic can undergo NAT. The forward hash key can be any combination of SIP, DIP, and protocol, and the reverse hash key is ignored.

NOTE: The Junos OS AMS configuration supports IPv4 and IPv6 traffic.

IPv6 Traffic on AMS Interfaces Overview

Starting in Junos OS release 14.2R1, you can use AMS interfaces for IPv6 traffic. To configure IPv6 support for an AMS interface, include the `family inet6` statement at the `[edit interfaces` *`ams-interface-name`* `unit 1]` hierarchy level. When `family inet` and `family inet6` are set for an AMS interface subunit, the `hash-keys` is configured at service-set level for interface style and at IFL level for next-hop style.

When a member interface of an AMS bundle fails, traffic destined to the failed member is redistributed among the remaining active members. The traffic (flows or sessions) traversing through the existing active members is unaffected. If M members are currently active, the expected result is that only about 1/M fraction of the traffic (flows/sessions) is impacted because that amount of traffic is shifted from the failed member to remain active members. When the failed member interface comes back online, only a fraction of the traffic is redistributed to the new member. If N members are currently active, the expected result is that only about 1/(N+1) fraction of the traffic (flows/sessions) is impacted because that amount of traffic moves to the new restored member. The 1/M and 1/(N+1) values assume that the flows are uniformly distributed among members, because a packet-hash is used to load-balance and because traffic usually contains a typical random combination of IP addresses (or any other fields that are used as load-balancing keys).

Similar to IPv4 traffic, for IPv6 packets, an AMS bundle must contain members of only one services PIC type. Separate AMS bundles on the same router can contain members of different services PIC types (for example, two MS-MICs in ams0, and two MS-MPC PICs in ams1).

The number of flows distributed, in an ideal environment, can be 1/N in a best-case scenario when the Nth member goes up or down. However, this assumption considers that the hash keys load-balance the real or dynamic traffic. For example, consider a real-world deployment where member A is serving only one flow, whereas member B is serving 10 flows. If member B goes down, then the number of flows disrupted is 10/11. The NAT pool-split behavior is designed to utilize the benefits of the rehash-minimization feature. The splitting of a NAT pool is performed for dynamic NAT scenarios (dynamic NAT, NAT64, and NAPT44).

If the original and redistributed flows are defined as follows:

- Member-original-flows—The traffic mapped to a member when all members are up.
- Member-redistributed-flows—The additional traffic mapped to a member when some other member fails. These traffic flows might need to be rebalanced when member interfaces come up and go down.

With the preceding definitions of the original and redistributed flows for member interfaces, the following observations apply:

- The member-original-flows of a member stay intact as long as that member is up. Such flows are not impacted when other members move between the up and down states.
- The member-redistributed-flows of a member can change when other members go up or down. This change of flows occurs because these additional flows need to be rebalanced among all active members. Therefore, the member-redistributed-flow can vary a lot based on other members going down or up. Although it might seem that when a member goes down, the flows on active-members are preserved, and that when a member goes up, flows on active-members are not preserved in an effective way, this behavior is only because of static or hash-based rebalancing of traffic among active members.

The rehash-minimization feature handles the operational changes in a member interface status only (such as member offline or member Junos OS reset). It does not handle changes in configuration. For example, addition or deletion, or activation and deactivation, of member interfaces at the `[edit interfaces ams`*N* `load-balancing-options member-interface mams-a/b/0]` hierarchy level requires the member PICs to be bounced. Twice NAT or hairpinning is not supported, similar to IPv4 support for AMS interfaces.

Member Failure Options and High Availability Settings

Because multiple service interfaces are configured as part of an AMS bundle, AMS configuration also provides for failover and high availability support. You can either configure one of the member interfaces as a backup interface that becomes active when any one of the other member interfaces goes down, or configure the AMS in such a way that when one of the member interfaces goes down, the traffic assigned to that interface is shared across the active interfaces.

The `member-failure-options` *configuration statement* enables you to configure how to handle traffic when a member interface fails. One option is to redistribute the traffic immediately among the other member interfaces. However, redistribution of traffic involves recalculating the hash tags, and might cause some disruption in traffic on all the member interfaces.

The other option is to configure the AMS to drop all traffic that is assigned to the failed member interface. With this you can optionally configure an interval, `rejoin-timeout`, for the AMS to wait for the failed interface to come back online after which the AMS can redistribute the traffic among other member interfaces. If the failed member interface comes back online before the configured wait time, traffic continues unaffected on all member interfaces, including the interface that has come back online and resumed the operations.

You can also control the rejoining of the failed interface when it comes back online. If you do not include the `enable-rejoin` statement in the `member-failure-options` configuration, the failed interface cannot rejoin the AMS when it comes back online. In such cases, you can manually rejoin that to the AMS by executing the `request interfaces revert` *`interface-name`* *operational mode command*.

The `rejoin-timeout` and `enable-rejoin` statements enable you to minimize traffic disruptions when member interfaces flap.

NOTE: When `member-failure-options` are not configured, the default behavior is to drop member traffic with a rejoin timeout of 120 seconds.

The `high-availability-options` configuration enables you to designate one of the member interfaces as a backup interface. The backup interface does not participate in routing operations as long as it remains a backup interface. When a member interface fails, the backup interface handles the traffic assigned to the failed interface. When the failed interface comes back online, it becomes the new backup interface.

In a many-to-one configuration (N:1), a single backup interface supports all other member interfaces in the group. If any of the member interfaces fails, the backup interface takes over. In this stateless configuration, data is not synchronized between the backup interface and the other member interfaces.

Starting in Junos OS Release 16.1, in a one-to-one configuration, a single active interface is paired with a single backup interface. If the active interface fails, the backup interface does take over. Configurations using `member-failure-options` are not available for one-to-one (1:1) high availability configurations.

When both `member-failure-options` and `high-availability-options` are configured for an AMS, the `high-availability-options` configuration takes precedence over the `member-failure-options` configuration. If a second failure occurs before the failed interface comes back online to be the new backup, the `member-failure-options` configuration takes effect.

Warm Standby Redundancy

Starting in Junos OS Release 17.2R1, you can use the same services interface as the backup in multiple AMS interfaces, resulting in an N:1 warm standby option for MS-MPCs and MS-MICs.

Each warm standby AMS interface contains two members; one member is the service interface you want to protect, called the primary interface, and one member is the secondary (backup) interface. The primary interface is the active interface and the backup interface does not handle any traffic unless the primary interface fails.

To configure warm standby on an AMS interface, you use the `redundancy-options` statement. You cannot use the `load-balancing-options` statement in a warm standby AMS interface.

To switch from the primary interface to the secondary interface, issue the `request interface switchover ams`*`N`* command.

To revert to the primary interface from the secondary interface, issue the `request interface revert ams`*`N`* command.

Configuring Aggregated Multiservices Interfaces

The aggregated multiservices (AMS) interface configuration in Junos OS enables you to combine services interfaces from multiple PICs to create a bundle of interfaces that can function as a single interface. You identify the PIC that you want to act as the backup.

1. Create an aggregated multiservices interface and add member interfaces. Starting in Junos OS Release 19.3R2, an MX-SPC3 Next Gen Services AMS interface can have up to 14 member interfaces with a maximum of 7 MX-SPC3 services cards with up to 2 PICs on each card. Starting with Junos OS Release 16.2, an MS-MPC AMS interface can have up to 36 member interfaces. In Junos OS Release 16.1 and earlier, an AMS interface can have a maximum of 24 member interfaces.

NOTE: The member interface format is mams-*a*/*b*/0, where *a* is the Flexible PIC Concentrator (FPC) slot number and *b* is the PIC slot number.

```
[edit interfaces]
user@host# set interface-name load-balancing-options member-interface mams-a/b/0
user@host# set interface-name load-balancing-options member-interface mams-a/b/0
```

For example on an MS-MPC, which can have up to four PICs:

```
[edit interfaces]
user@host# set ams1 load-balancing-options member-interface mams-1/1/0
user@host# set ams1 load-balancing-options member-interface mams-1/2/0
```

For example on an MX-SPC3, which can have up to two PICs:

```
[edit interfaces]
user@host# set ams1 load-balancing-options member-interface mams-1/0/0
user@host# set ams1 load-balancing-options member-interface mams-1/1/0
```

2. Configure logical units for the AMS interface.

```
[edit interfaces]
user@host# set interface-name unit logical-unit-number family family
user@host# set interface-name unit logical-unit-number family family
```

For example:

```
[edit interfaces]
user@host# set ams1 unit 1 family inet
user@host# set ams1 unit 2 family inet6
```

3. Configure member failure options.

```
[edit interfaces interface-name]
user@host# set load-balancing-options member-failure-options drop-member-traffic rejoin-
```

```
timeout seconds
user@host# set load-balancing-options member-failure-options drop-member-traffic enable-rejoin
```

For example:

```
[edit interfaces ams1]
user@host# set load-balancing-options member-failure-options drop-member-traffic rejoin-
timeout 1000
user@host# set load-balancing-options member-failure-options drop-member-traffic enable-rejoin
```

4. Configure the preferred backup.

```
[edit interfaces interface-name]
user@host# set load-balancing-options high-availability-options many-to-one preferred-backup
preferred-backup
```

For example:

```
[edit interfaces ams1]
user@host# set load-balancing-options high-availability-options many-to-one preferred-backup
mams-1/2/0
```

5.

NOTE: This step is not applicable to the Next Gen Services MX-SPC3 services card in the MX240, MX480 or MX960 chassis.

If the AMS interface has more than 24 member interfaces, set the service PIC boot timeout value to 240 or 300 seconds for every services PIC on the MX Series router. We recommend that you use a value of 240.

NOTE: Starting with Junos OS Release 16.2, an AMS interface can have up to 36 member interfaces. In Junos OS Release 16.1 and earlier, an AMS interface could have a maximum of 24 member interfaces.

```
[edit interfaces interface-name multiservice-options]
user@host# set pic-boot-timeout (240 | 300);
```

For example:

```
[edit interfaces sp-1/1/0 multiservice-options]
user@host# set pic-boot-timeout 240
```

SEE ALSO

Understanding Aggregated Multiservices Interfaces for Next Gen Services

Configuring Load Balancing on AMS Infrastructure

IN THIS SECTION

- Configuring AMS Infrastructure | 1020
- Configuring High Availability | 1022
- Load Balancing Network Address Translation Flows | 1023

Configuring load balancing requires an aggregated multiservices (AMS) system. AMS involves grouping several services PICs together. An AMS configuration eliminates the need for separate routers within a system. The primary benefit of having an AMS configuration is the ability to support load balancing of traffic across multiple services PICs.

AMS is supported on the MS-MPC and MS-MIC. Starting in Junos OS Release 19.3R2, AMS interfaces are supported on the MX-SPC3.

High availability (HA) is supported on AMS infrastructure on all MX Series 5G Universal Routing Platforms. AMS has several benefits:

- Support for configuring behavior if a services PIC that is part of the AMS configuration fails
- Support for specifying hash keys for each service set in either direction
- Support for adding routes to individual PICs within the AMS system

Configuring AMS Infrastructure

AMS supports load balancing across multiple service sets. All ingress or egress traffic for a service set can be load balanced across different services PICs. To enable load balancing, you have to configure an aggregate interface with existing services interfaces.

To configure failure behavior in AMS, include the `member-failure-options` statement:

```
[edit interfaces ams1]
load-balancing-options {
    member-failure-options {
        drop-member-traffic {
            rejoin-timeout rejoin-timeout;
        }
        redistribute-all-traffic {
            enable-rejoin;
        }
    }
}
```

If a PIC fails, you can configure the traffic to the failed PIC to be redistributed by using the `redistribute-all-traffic` statement at the `[edit interfaces` *`interface-name`* `load-balancing-options member-failure-options]` hierarchy level. If the `drop-member-traffic` statement is used, all traffic to the failed PIC is dropped. Both options are mutually exclusive.

NOTE: If `member-failure-options` is not explicitly configured, the default behavior is to drop member traffic with a rejoin timeout of 120 seconds.

Only mams- interfaces (services interfaces that are part of AMS) can be aggregated. After an AMS interface has been configured, you cannot configure the individual constituent mams- interfaces. A mams- interface cannot be used as an ams interface (this is not applicable to Next Gen Services MX-SPC3). AMS supports IPv4 (`family inet`) and IPv6 (`family inet6`). You cannot configure addresses on an AMS interface. Network Address Translation (NAT) is the only application that runs on AMS infrastructure at this time.

NOTE: You cannot configure unit 0 on an AMS interface.

To support multiple applications and different types of translation, AMS infrastructure supports configuring hashing for each service set. You can configure the hash keys separately for ingress and egress. The default configuration uses source IP, destination IP, and the protocol for hashing; incoming-interface for ingress and outgoing-interface for egress are also available.

NOTE: When using AMS in a load-balanced setup for the NAT solution, the number of NAT IP addresses must be greater than or equal to the number of active mams-interfaces you have added to the AMS bundle.

Configuring High Availability

In an AMS system configured with high availability, a designated services PIC acts as a backup for other active PICs that are part of the AMS system in a many-to-one (N:1) backup configuration. In a N:1 backup configuration, one PIC is available as backup for all other active PICs. If any of the active PICs fail, the backup PIC takes over for the failed PIC. In an N:1 (stateless) backup configuration, traffic states and data structures are not synchronized between the active PICs and the backup PIC.

An AMS system also supports a one-to-one (1:1) configuration. In the case of 1:1 backup, a backup interface is paired with a single active interface. If the active interface fails, the backup interface takes over. In a 1:1 (stateful) configuration, traffic states and data structures are synchronized between the active PICs and the backup PIC. Stateful synchronization is required for high availability of IPsec connections. For IPsec connections, AMS supports 1:1 configuration only.

NOTE: IPsec connections are not supported on the MX-SPC3 in this release.

High availability for load balancing is configured by adding the `high-availability-options` statement at the `[edit interfaces` *`interface-name`* `load-balancing-options]` hierarchy level.

To configure N:1 high availability, include the `high-availability-options` statement with the `many-to-one` option:

```
[edit interfaces ams1]
load-balancing-options {
    high-availability-options {
        many-to-one {
            preferred-backup preferred-backup;
        }
    }
}
```

Starting in Junos OS Release 16.1, you can configure stateful 1:1 high availability on an MS-MPC. To configure stateful 1:1 high availability, at the `[edit interfaces` *`interface-name`* `load-balancing-options]` hierarchy level, include the `high-availability-options` statement with the `one-to-one` option:

NOTE: The Next Gen Services MX-SPC3 services card does not support AMS 1:1 high availability.

```
[edit interfaces ams1]
load-balancing-options {
    high-availability-options {
        one-to-one {
            preferred-backup preferred-backup;
        }
    }
}
```

Load Balancing Network Address Translation Flows

Network Address Translation (NAT) has been programmed as a plug-in and is a function of load balancing and high availability. The plug-in runs on AMS infrastructure. All flows for translation are automatically distributed to different services PICs that are part of the AMS infrastructure. In case of failure of an active services PIC, the configured backup PIC takes over the NAT pool resources of the failed PIC. The hashing method selected depends on the type of NAT. Using NAT on AMS infrastructure has a few limitations:

- NAT flows to failed PICs cannot be restored.
- There is no support for IPv6 flows.

  IPv6 address pools are not supported with AMS, however NAT64 is supported with AMS, so that IPv6 flows enters AMS.

  NAT64 is supported for Next Gen Services on the MX-SPC3 services card, there is no support of NAT66. IPv6 flows for different NAT services are supported except where the translation is required to be IPv6 to IPv6 or IPv4 to IPv6.
- Twice NAT is not supported for load balancing on MS-MPC cards.

  Twice NAT is supported for load balancing on the Next Gen Services MX-SPC3 services card.
- Deterministic NAT uses warm-standby AMS configuration and can distribute the load using multiple AMS bundles in warm-standby mode.

Configuring Warm Standby for Services Interfaces

You can configure an N:1 warm standby option for MS-MPCs, MS-MICs, and MX-SPC3s by creating multiple aggregated multiservices (AMS) interfaces, each of which contains the service interface you want to backup and the service interface that acts as the backup. The same backup service interface can be used in all these AMS interfaces. Starting in Junos OS Release 19.3R2, the N:1 warm standby option is supported on the MX-SPC3.

To configure warm standby for services interfaces:

1. Create an AMS interface.

   ```
   [edit interfaces]
   user@host# set amsN
   ```

   The variable *N* is a unique number, such as 0 or 1.
2. Specify the primary service interface that you want to backup.

   ```
   [edit interfaces amsN]
   user@host# set redundancy-options primary mams-a/b/0
   ```

   The variable *a* is the FPC slot number and *b* is the PIC slot number for the primary service interface.
3. Specify the secondary service interface, which backs up the primary interface.

   ```
   [edit interfaces amsN]
   user@host# set redundancy-options secondary mams-a/b/0
   ```

   The variable *a* is the FPC slot number and *b* is the PIC slot number for the secondary service interface.
4. Repeat Steps 1 through 3 to create an AMS interface for each service interface that you want to backup. You can use the same secondary service interface in each AMS interface.

SEE ALSO

Understanding Aggregated Multiservices Interfaces | **1010**

Example: Configuring an Aggregated Multiservices Interface (AMS)

IN THIS SECTION

- Hardware and Software Requirements | 1025
- Overview | 1025
- Configuration | 1026
- Verification | 1031

Hardware and Software Requirements

This example requires MX Series routers that have services interfaces installed in that and Junos OS Release 13.2 running on that.

Overview

The aggregated multiservices (AMS) interface configuration in Junos OS enables you to combine multiple services interfaces to create a bundle of interfaces that can function as a single interface. This example shows you how to configure an AMS interface, load-balancing options, member failure options, high availability settings on an AMS interface, and an interface-style service set configuration that uses the AMS interface.

> **NOTE**: You cannot include MS-DPCs or other multiservices PICs in an AMS configuration that contains MS-MICs or MS-MPCs as member interfaces.

An MS-PIC contains only one interface, whereas the MS-MPC contains four interfaces. To utilize the entire MS-MPC in a single AMS bundle, all the four member interfaces need to be assigned to that AMS bundle.

Keep the following points in mind for every member interface (XLP chip) needs to be part of the AMS interface bundle:

- XLP-based line cards from the same MPC can be part of multiple AMS bundles.
- Multiple XLP chips from several MPCs can also be part of a single bundle (up to eight member interfaces in an AMS bundle, depending on the deployment requirement).
- It is not necessary that all the XLP chips from the same MS-MPC must be part of the same AMS bundle. Some of the XLP chips can be part of an AMS bundle, while other XLP chips can be

standalone `ms-` interfaces or need not be configured. However, the same XLP chip cannot be part of two different AMS interfaces at the same time. For example, each XLP chip from the same MS-MPC can be grouped into four different AMS bundles, based on the deployment needs.

- A maximum of up to eight member interfaces can be assigned to an AMS bundle.

For more information about AMS interfaces, see "Understanding Aggregated Multiservices Interfaces" on page 1010 .

Configuration

IN THIS SECTION

- Procedure | 1026

Procedure

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the [edit] hierarchy level.

Adding Member Interfaces

```
set interfaces ams0 load-balancing-options member-interface mams-0/0/0
set interfaces ams0 load-balancing-options member-interface mams-0/1/0
set interfaces ams0 load-balancing-options member-interface mams-1/0/0
set interfaces ams0 load-balancing-options member-interface mams-1/1/0
set interfaces ams0 load-balancing-options member-interface mams-2/0/0
set interfaces ams0 load-balancing-options member-interface mams-2/1/0
```

Configuring Logical Units

```
set interfaces ams0 unit 1 family inet
```

Configuring Member Failure Options

```
set interfaces ams0 load-balancing-options member-failure-options drop-member-traffic rejoin-timeout 300
set interfaces ams0 load-balancing-options member-failure-options drop-member-traffic enable-rejoin
```

Configuring High Availability Options

```
set interfaces ams0 load-balancing-options high-availability-options many-to-one preferred-backup mams-1/0/0
```

Configuring Service Set and Interface Services

```
set services service-set ams-ss1 interface-service service-interface ams0.1
set services service-set ams-ss1 interface-service load-balancing-options hash-keys ingress-key source-ip
set services service-set ams-ss1 interface-service load-balancing-options hash-keys egress-key destination-ip
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

1. Create an aggregated multiservices interface and add member interfaces.

   NOTE: You cannot configure the same mams to be part of two different AMS interfaces at the same time.

   ```
   [edit]
   user@router1# set interfaces ams0 load-balancing-options member-interface mams-0/0/0
   user@router1# set interfaces ams0 load-balancing-options member-interface mams-0/1/0
   user@router1# set interfaces ams0 load-balancing-options member-interface mams-1/0/0
   user@router1# set interfaces ams0 load-balancing-options member-interface mams-1/1/0
   ```

```
user@router1# set interfaces ams0 load-balancing-options member-interface mams-2/0/0
user@router1# set interfaces ams0 load-balancing-options member-interface mams-2/1/0
```

2. Configure logical units for the AMS interface.

   NOTE: An AMS interface and its member interfaces cannot share the same logical interface units. For example, if one of the member interfaces has logical units 1 and 2 configured on it, you cannot configure logical units 1 and 2 for the AMS. Similarly, if you have configured logical units 3 and 4 on the AMS, you cannot configure those units on any of the member interfaces.

   ```
   [edit interfaces]
   user@router1# set ams0 unit 1 family inet
   ```

3. Configure member failure options.

   ```
   [edit interfaces ams0]
   user@router1# set load-balancing-options member-failure-options drop-member-traffic rejoin-timeout 300
   user@router1# set load-balancing-options member-failure-options drop-member-traffic enable-rejoin
   ```

   NOTE: This example shows the `drop-member-traffic` configuration. However, you if you would like to redistribute the traffic to other available members when one of the member interfaces goes down, you can include the `redistribute-all-traffic` statement instead of the `drop-member-traffic` statement.

   The default behavior, when the `member-failure-options` configuration is not included, is to drop member traffic with a rejoin timeout of 120 seconds.

4. Configure the high-availability options.

   ```
   [edit interfaces ams0]
   user@router1# set load-balancing-options high-availability-options many-to-one preferred-backup mams-1/0/0
   ```

5. Configure interface style services.

```
[edit services]
user@router1# set service-set ams-ss1 interface-service service-interface ams0.1
user@router1# set service-set ams-ss1 interface-service load-balancing-options hash-keys
ingress-key source-ip
user@router1# set service-set ams-ss1 interface-service load-balancing-options hash-keys
egress-key destination-ip
```

6. If you are done configuring the device, commit the configuration.

```
[edit]
user@router1# commit
```

Table 31: Key Configuration Statements Used in this Example

Statement	Description
member-interface	Adds a member interface (mams) to the AMS bundle.
drop-member-traffic	Specifies that all traffic to a member be dropped in case the member interface fails.
rejoin-timeout	Specifies the time interval, in seconds, for the AMS to wait before declaring a member interface down. If the failed member comes back online during this period, it can rejoin the AMS and resume traffic forwarding. The range is 0 through 1000 seconds.
enable-rejoin	Specifies whether a failed interface be allowed to rejoin the AMS when it comes back online. If this statement is not included in the configuration, you must manually add the interface to the AMS when the interface is back online.
preferred-backup	Designates a member interface as the floating backup.

Table 31: Key Configuration Statements Used in this Example *(Continued)*

Statement	Description
`interface-services`	Specifies a service interface, an AMS interface in this example, to handle interface services.
`hash-keys`	Specifies the load-balancing hash keys. You can configure the following hash key values: `source-ip`, `destination-ip`, `iif` (incoming interface), `oif` (outgoing interface), and `protocol`. **NOTE**: For services that require traffic symmetry, you must configure symmetrical hashing. Symmetrical hashing configuration ensures that both forward and reverse traffic are routed through the same member interface.

Results

From the configuration mode, confirm your configuration by entering the `show interfaces ams0` command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@router1# show interfaces ams0
load-balancing-options {
    member-interface mams-0/0/0;
    member-interface mams-0/1/0;
    member-interface mams-1/0/0;
    member-interface mams-1/1/0;
    member-interface mams-2/0/0;
    member-interface mams-2/1/0;
    member-failure-options {
        drop-member-traffic {
            rejoin-timeout 300;
            enable-rejoin;
        }
    }
    high-availability-options {
        many-to-one {
            preferred-backup mams-1/0/0;
        }
    }
}
    unit 1 {
        family inet;
    }
```

```
user@router1# show services
service-set ams-ss1 {
    interface-service {
        service-interface ams0.1;
        load-balancing-options {
            hash-keys {
                ingress-key source-ip;
                egress-key destination-ip;
            }
        }
    }
}
```

Verification

IN THIS SECTION

- Verifying the AMS Configuration | 1031

Confirm that the configuration is working properly.

Verifying the AMS Configuration

Purpose

Verify the AMS configuration and status of member interfaces.

Action

From operational mode, enter the `show` command.

```
user@router1> show interfaces load-balancing detail
Load-balancing interfaces detail
Interface         : ams0
  State           : Up
  Last change     : 00:01:28
  Member count    : 6
  HA Model        : Many-to-One
  Members         :
      Interface     Weight   State
      mams-0/0/0    10       Active
      mams-0/1/0    10       Active
      mams-1/0/0    10       Backup
      mams-1/1/0    10       Active
      mams-2/0/0    10       Active
      mams-2/1/0    10       Active
```

Meaning

Shows that `ams0` has six member interfaces with a many-to-one backup configuration. Of the six member interfaces, five are in active state and one, mams-1/0/0, is in backup state.

SEE ALSO

Understanding Aggregated Multiservices Interfaces | 1010

Example: Configuring Next-Hop Style Services on an Aggregated Multiservices Interface

IN THIS SECTION

- Hardware and Software Requirements | 1037
- Overview | 1037

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the [edit] hierarchy level.

Configuring an aggregated multiservices interface

```
set interfaces ams0 load-balancing-options member-interface mams-1/0/0
set interfaces ams0 load-balancing-options member-interface mams-1/1/0
set interfaces ams0 load-balancing-options member-interface mams-2/0/0
set interfaces ams0 load-balancing-options member-interface mams-2/1/0
set interfaces ams0 unit 1 family inet
set interfaces ams0 unit 1  service-domain inside
set interfaces ams0 unit 2 family inet
set interfaces ams0 unit 2 service-domain outside
```

Configuring Routing Instances that Use AMS interfaces

```
set routing-instances ri-internal instance-type virtual-router
set routing-instances ri-internal interface ge-0/0/2.0
set routing-instances ri-internal interface ams0.1
set routing-instances ri-internal routing-options static route 22.22.22.0/24 next-hop ams0.1
set routing-instances ri-external instance-type virtual-router
set routing-instances ri-external interface ge-2/0/6.0
set routing-instances ri-external interface ams0.2
set routing-instances ri-external routing-options static route 0.0.0.0/0 next-hop ams0.2
```

Configuring Hash Keys

```
set interfaces ams0 unit 1 load-balancing-options hash-keys ingress-key source-ip protocol
set interfaces ams0 unit 2 load-balancing-options hash-keys ingress-key destination-ip protocol
```

Configure Next Hop Services

```
set services service-set ams-test stateful-firewall-rules sfw1
set services service-set ams-test next-hop-service inside-service-interface ams0.1
set services service-set ams-test next-hop-service outside-service-interface ams0.2
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see “*Using the CLI Editor in Configuration Mode*” in the *CLI User Guide*.

1. Configure an aggregated multiservices interface and the load-balancing options.

```
[edit interfaces ams0]
user@router1# set load-balancing-options member-interface mams-1/0/0
user@router1# set load-balancing-options member-interface mams-1/1/0
user@router1# set load-balancing-options member-interface mams-2/0/0
user@router1# set load-balancing-options member-interface mams-2/1/0
user@router1# set unit 1 family inet
user@router1# set unit 1 service-domain inside
user@router1# set unit 2 family inet
user@router1# set unit 2 service-domain outside
```

2. Configure routing instances that use the aggregated multiservices interfaces configured in the first step.

```
[edit routing-instances]
user@router1# set ri-internal instance-type virtual-router
user@router1# set ri-internal interface ge-0/0/2.0
user@router1# set ri-internal interface ams0.1
user@router1# set ri-internal routing-options static route 22.22.22.0/24 next-hop ams0.1
user@router1# set ri-external instance-type virtual-router
user@router1# set ri-external interface ge-2/0/6.0
user@router1# set ri-external interface ams0.2
user@router1# set ri-external routing-options static route 0.0.0.0/0 next-hop ams0.2
```

3. Configure hash keys for the aggregated multiservices interfaces.

NOTE: Unlike in the interface-style configuration where hash keys are defined in the service-set configuration, for next-hop services, the hash keys are specified in the AMS configuration under the logical units.

```
[edit interfaces ams0]
user@router1# set unit 1 load-balancing-options hash-keys ingress-key source-ip protocol
user@router1# set unit 2 load-balancing-options hash-keys ingress-key destination-ip protocol
```

4. Configure next-hop style services under the service-set configuration.

```
[edit services service-set ams-test]
user@router1# set stateful-firewall-rules sfw1
user@router1# set next-hop-service inside-service-interface ams0.1
user@router1# set next-hop-service outside-service-interface ams0.2
```

5. Commit the configuration.

```
[edit]
user@router1# commit
```

Results

From the configuration mode, confirm your configuration by entering the `show interfaces ams0`, `show routing-instances`, and `show services service-set ams-test` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@router1# show interfaces ams0
load-balancing-options {
    member-interface mams-1/0/0;
    member-interface mams-1/1/0;
    member-interface mams-2/0/0;
    member-interface mams-2/1/0;
    member-failure-options {
        redistribute-all-traffic {
            enable-rejoin;
        }
    }
}
    unit 1 {
        family inet;
        service-domain inside;
        load-balancing-options {
            hash-keys {
                ingress-key [ source-ip protocol ];
            }
        }

    }
    unit 2 {
        family inet;
        service-domain outside;
        load-balancing-options {
            hash-keys {
                ingress-key [  destination-ip protocol ];
            }
        }
    }
```

```
user@router1# **show routing-instances**
ri-internal {
    instance-type virtual-router;
    interface ge-0/0/2.0;
    interface ams0.1
    routing-options {
        static {
            route 22.22.22.0/24 next-hop ams0.1;
        }
    }
}
    ri-external {
        instance-type virtual-router;
        interface ge-2/0/6.0;
        interface ams0.2
        routing-options {
            static {
                route 0.0.0.0/0 next-hop ams0.2;
            }
        }
    }
```

```
user@router1# show services service-set ams
stateful-firewall-rules sfw1;
next-hop-service {
    inside-service-interface ams0.1;
    outside-service-interface ams0.2;
}
```

Hardware and Software Requirements

MX Series routers with services interfaces installed and running Junos OS Release 13.2.

Overview

Starting with Release 13.2, Junos OS extends next-hop style services support to aggregated multiservices (AMS) interfaces. In releases earlier than 12.3, only interface style services configurations were supported on AMS interfaces.

The next-hop style services configuration on AMS interfaces is different from the interface style services configuration. For next-hop style services, the load-balancing hash keys are defined as part of the logical unit configuration of the AMS interface. For interface style services, the hash keys configuration falls under the service-set configuration.

This example explains the next-hop style services configuration on an AMS interface, and shows the verification steps to verify that the configuration is working correctly.

SEE ALSO

Understanding Aggregated Multiservices Interfaces | 1010

Example: Configuring Static Source Translation on AMS Infrastructure

This example shows a static source translation configured on an AMS interface. The flows will be load balanced across member interfaces with this example.

Configure the AMS interface ams0 with load balancing options.

```
[edit interfaces ams0]
```

```
load-balancing-options {
member-interface mams-5/0/0;
member-interface mams-5/1/0;
}
unit 1 {
    family inet;
}
unit 2 {
    family inet;
}
```

Configure hashing for the service set for both ingress and egress traffic.

```
[edit services service-set ss1]
interface-service {
    service-interface ams0.1;
    load-balancing-options {
        hash-keys {
            ingress-key destination-ip;
            egress-key source-ip;
        }
    }
}
```

NOTE: Hashing is determined based on whether the service set is applied on the ingress or egress interface.

Configure two NAT pools because you have configured two member interfaces for the AMS interface.

```
[edit services]
nat {
    pool p1 {
        address-range low 20.1.1.80 high 20.1.1.80;
    }
pool p2 {
        address 20.1.1.81/32;
    }
}
```

Configure the NAT rule and translation.

```
[edit services]
nat {
    rule r1 {
        match-direction input;
        term t1 {
            from {
                source-address {
                    20.1.1.2/32;
                }
            }
            then {
                translated {
                    source-pool p1;
                    translation-type {
                        basic-nat44;
                    }
                }
           }
              term t1 {
                 from {
                  source-address {
                      40.1.1.2/32;
                  }
             }
             then {
                 translated {
                     source-pool p2;
                     translation-type {
                         basic-nat44;
                    }
                }
           }
      }
 }
```

NOTE: A similar configuration can be applied for translation types `dynamic-nat44` and `napt-44`. Twice NAT cannot run on AMS infrastructure at this time.

SEE ALSO

Configuring Load Balancing on AMS Infrastructure | **1020**

Understanding Aggregated Multiservices Interfaces | **1010**

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
19.3R2	Starting in Junos OS Release 19.3R2, an MX-SPC3 Next Gen Services AMS interface can have up to 16 member interfaces with a maximum of 8 MX-SPC3 services cards with up to 2 PICs on each card.
19.3R2	Starting in Junos OS Release 19.3R2, AMS interfaces are supported with the MX-SPC3.
19.3R2	Starting in Junos OS Release 19.3R2, the N:1 warm standby option is also supported on the MX-SPC3 if you are running Next Gen Services.
19.2R1	Starting with Junos OS Release 19.2R1, you can use up to 60 PICs across different AMS bundles on a MX2020 router. The hard limit of maximum 36 member interfaces per AMS bundle still exists. However, in the chassis, there can be multiple AMS bundles such that 15 MS-MPCs can be configured across these bundles.
17.2R1	Starting in Junos OS Release 17.2R1, you can use the same services interface as the backup in multiple AMS interfaces, resulting in an N:1 warm standby option for MS-MPCs and MS-MICs.
17.1	Starting with Junos OS Release 17.1R1, AMS supports IPSec tunnel distribution for next-hop style service-sets. However, interface-style IPSec service sets are not supported.
16.2	Starting with Junos OS Release 16.2 (except Junos OS Release 17.3R3-S7), an AMS interface can have up to 36 member interfaces.
16.2	Starting with Junos OS Release 16.2, an MS-MPC AMS interface can have up to 36 member interfaces.

16.1	Starting in Junos OS Release 16.1, in a one-to-one configuration, a single active interface is paired with a single backup interface. If the active interface fails, the backup interface does take over.
16.1	Starting in Junos OS Release 16.1, you can configure stateful 1:1 high availability on an MS-MPC.
14.2	Starting in Junos OS release 14.2R1, you can use AMS interfaces for IPv6 traffic.
14.1	Starting in Junos OS Release 14.1R4, input interface aliases are not created for MS-MPCs and MS-MICs.

CHAPTER 40

Services Card Redundancy for Multiservices PIC

IN THIS CHAPTER

- Configuring AS or Multiservices PIC Redundancy | **1042**

Configuring AS or Multiservices PIC Redundancy

You can configure AS or Multiservices PIC redundancy on M Series and T Series routers, except TX Matrix routers, that have multiple AS or Multiservices PICs. To configure redundancy, you specify a redundancy services PIC (`rsp`) interface in which the primary PIC is active and a secondary PIC is on standby. If the primary PIC fails, the secondary PIC becomes active, and all service processing is transferred to it. If the primary AS or Multiservices PIC is restored, it remains on standby and does not preempt the secondary PIC; you need to manually restore the services to the primary PIC. To determine which PIC is currently active, issue the `show interfaces redundancy` command.

Failover to the secondary PIC occurs under the following conditions:

- The primary PIC, FPC, or Packet Forwarding Engine goes down, resets, or is physically removed from the router.

- The PIC or FPC is taken offline using the `request chassis pic fpc-slot` *`slot-number`* `pic-slot` *`slot-number`* `offline` or `request chassis fpc slot` *`slot-number`* `offline` command. For more information, see the CLI Explorer.

- The driver watchdog timer expires.

- The `request interface switchover` command is issued. For more information, see the CLI Explorer.

NOTE: Adaptive Services and Multiservices PICs in Layer-2 mode (running Layer 2 services) are not rebooted when a MAC flow-control situation is detected.

> **NOTE**: When you perform a switchover from a primary PIC to a secondary or standby PIC or a revert operation by issuing request interfaces (revert | switchover) command for redundancy services PICs (rsp), the PIC that was previously the active PIC before the switchover or reversion is automatically rebooted. The reboot of the PIC that was previously active and functioning as the primary PIC does not disrupt traffic forwarding.

The physical interface type rsp specifies the pairings between primary and secondary sp interfaces to enable redundancy. To configure an AS or Multiservices PIC as the backup, include the redundancy-options statement at the [edit interfaces rsp*number*] hierarchy level:

```
[edit interfaces rspnumber]
redundancy-options {
    primary sp-fpc/pic/port;
    secondary sp-fpc/pic/port;
    hot-standby;
}
```

For the rsp interface, *number* can be from 0 through 15.

> **NOTE**: You can include a similar redundancy configuration for Link Services IQ (LSQ) PICs at the [edit interfaces rlsq*number*] hierarchy level. For more information, see "Configuring LSQ Interface Redundancy in a Single Router Using Virtual Interfaces" on page 921 .

The following constraints apply to redundant AS or Multiservices PIC configurations:

- The services supported in redundancy configurations include stateful firewall, NAT, IDS, and IPsec. Services mounted on the AS or Multiservices PIC that use interface types other than sp- interfaces, such as tunneling and voice services, are not supported. For information on flow monitoring redundancy, see *Configuring Services Interface Redundancy with Flow Monitoring*.

  > **NOTE**: For IPsec functionality, the router no longer needs to renegotiate security associations (SAs) during warm standby PIC switchover. Instead, the warm standby feature has been made stateful by periodically setting a checkpoint between the working state of the PIC and the Routing Engine, which should lessen the downtime during switchover. If you prefer to retain the earlier behavior, you can include the clear-ipsec-sas-on-pic-restart statement at the [edit services ipsec-vpn] hierarchy level. If you enable this capability, the router renegotiates the

IPsec SAs on warm standby PIC switchover. For more information, see "Configuring Security Associations" on page 645 .

- We recommend that you pair the same model type in RSP configurations, such as two ASMs or two AS2 PICs. If you pair unlike models, the two PICs may perform differently.
- You can specify an AS or Multiservices PIC (`sp` interface) as the primary for only one `rsp` interface.
- An `sp` interface can be a secondary for multiple `rsp` interfaces. However, the same `sp` interface cannot be configured as a primary interface in one `rsp` configuration and as a secondary in another configuration.
- When the secondary PIC is active, if another primary PIC that is paired with it in an `rsp` configuration fails, no failover takes place.
- When you configure an AS or Multiservices PIC within a redundant configuration, the `sp` interface cannot have any configured services. Apply the configurations at the `[edit interfaces rsp`*`number`*`]` hierarchy level, using, for example, the `unit` and `services-options` statements. Exceptions include the `multiservice-options` statement used in flow monitoring configurations, which can be configured separately for the primary and secondary `sp` interfaces, and the `traceoptions` statement.
- All the operational mode commands that apply to `sp` interfaces also apply to `rsp` interfaces. You can issue `show` commands for the `rsp` interface or the primary and secondary `sp` interfaces.
- If a secondary PIC fails while it is in use, the `rsp` interface returns to the “ not present” state. If the primary PIC comes up later, service is restored to it.
- For redundant Multiservices (rms-) interfaces, similar to the configuration of other bundle interfaces, the properties of the Multiservices (ms-) member interfaces, such as the logical unit and the address family, are inherited from the underlying rms- interface. If you previously configured the member ms- interface properties separately, and attempt to configure the rms- interface properties by using the relevant statements at the `[edit interfaces rms`*`number`*`]` hierarchy level, an error occurs when you perform a commit check operation. You must configure the properties of interfaces that are part of the rms- interface only by using the statements at the `[edit interfaces rms`*`number`*`]` hierarchy level.

RELATED DOCUMENTATION

Services PICs-Overview

Examples: Configuring Services Interfaces | **21**

Example: Configuring an Aggregated Multiservices Interface (AMS) | **1025**

13
PART

Voice Services

Voice Services | 1046

CHAPTER 41

Voice Services

IN THIS CHAPTER

- Voice Services | 1046

Voice Services

IN THIS SECTION

- Voice Services Overview | 1046
- Configuring Services Interfaces for Voice Services | 1047
- Configuring Encapsulation for Voice Services | 1050
- Configuring Network Interfaces for Voice Services | 1051
- Examples: Configuring Voice Services | 1052

Voice Services Overview

Adaptive services interfaces include a voice services feature that allows you to specify interface type `lsq-`*`fpc`*`/`*`pic`*`/`*`port`* to accommodate voice over IP (VoIP) traffic. This interface uses compressed RTP (CRTP), which is defined in RFC 2508, *Compressing IP/UDP/RTP Headers for Low-Speed Serial Links*.

CRTP enables VoIP traffic to use low-speed links more effectively, by compressing the 40-byte IP/UDP/RTP header down to 2 to 4 bytes in most cases.

Voice services on the AS and MultiServices PICs support single-link PPP-encapsulated IPv4 traffic over the following physical interface types: ATM2, DS3, E1, E3, OC3, OC12, STM1, and T1, including the channelized versions of these interfaces.

Voice services do not require a separate service rules configuration.

Voice services also support LFI on Juniper Networks M Series Multiservice Edge routers, except the M320 router. For more information about configuring voice services, see "Configuring Services Interfaces for Voice Services" on page 1047 .

For link services IQ interfaces (`lsq`) only, you can configure CRTP with multiclass MLPPP (MCML). MCML greatly simplifies packet ordering issues that occur when multiple links are used. Without MCML, all voice traffic belonging to a single flow is hashed to a single link in order to avoid packet ordering issues. With MCML, you can assign voice traffic to a high-priority class, and you can use multiple links. For more information about MCML support on link services IQ interfaces, see "Configuring Link Services and CoS on Services PICs" on page 856 .

Configuring Services Interfaces for Voice Services

IN THIS SECTION

- Configuring the Logical Interface Address for the MLPPP Bundle | **1048**
- Configuring Compression of Voice Traffic | **1048**
- Configuring Delay-Sensitive Packet Interleaving | **1049**
- Example: Configuring Compression of Voice Traffic | **1050**

You define voice service properties such as compression by configuring statements and values for a voice services interface, specified by the interface type `lsq-`. You can include the following statements:

```
encapsulation mlppp;
family inet {
    address address;
}
compression {
    rtp {
        f-max-period number;
        maximum-contexts number <force>;
        port {
            minimum port-number;
            maximum port-number;
        }
        queues [ queue-numbers ];
    }
}
fragment-threshold bytes;
```

You can include these statements at the following hierarchy levels:

- `[edit interfaces (lsq | ls)-fpc/pic/port unit logical-unit-number]`
- `[edit logical-systems logical-system-name interfaces (lsq | ls)-fpc/pic/port unit logical-unit-number]`

The following sections provide detailed instructions for configuring for voice services on services interfaces:

Configuring the Logical Interface Address for the MLPPP Bundle

To configure the logical address for the MLPPP bundle, include the `address` statement:

```
address address {
    ...
}
```

You can configure this statement at the following hierarchy levels:

- `[edit interfaces (lsq | ls)-fpc/pic/port unit logical-unit-number family inet]`
- `[edit logical-systems logical-system-name interfaces (lsq | ls)-fpc/pic/port unit logical-unit-number family inet]`

address specifies an IP address for the interface. AS and Multiservices PICs support only IP version 4 (IPv4) addresses, which are therefore configured under the `family inet` statement.

For information on other addressing properties you can configure that are not specific to service interfaces, see the Junos OS Network Interfaces Library for Routing Devices.

Configuring Compression of Voice Traffic

You can specify how a services interface handles voice traffic compression by including the `compression` statement:

```
compression {
    rtp {
        f-max-period number;
        maximum-contexts number <force>;
        port {
            minimum port-number;
            maximum port-number;
        }
        queues [ queue-numbers ];
    }
}
```

You can include this statement at the following hierarchy levels:

- [edit interfaces (lsq | ls)-*fpc/pic/port* unit *logical-unit-number*]
- [edit logical-systems *logical-system-name* interfaces (lsq | ls)-*fpc/pic/port* unit *logical-unit-number*]

The following statements configure the indicated compression properties:

- f-max-period *number*—Sets the maximum number of compressed packets to insert between the transmission of full headers. If you do not include the statement, the default is 255 packets.
- maximum-contexts *number* <force>—Specifies the maximum number of RTP contexts to accept during negotiation. The optional force statement requires the PIC to use the value specified for maximum RTP contexts, regardless of the negotiated value. This option enables interoperation with Junos OS Releases that base the RTP context value on link speed.
- port, minimum *port-number*, and maximum *port-number*—Specify the lower and upper boundaries for a range of UDP destination port values on which RTP compression takes effect. Values for *port-number* can range from 0 through 65,535. RTP compression is applied to traffic transiting the ports within the specified range.
- queues [*queue-numbers*]—Specifies one or more of queues q0, q1, q2, and q3 . RTP compression is applied to the traffic in the specified queues.

> **NOTE**: If you specify both a port range and one or more queues, compression takes place if either condition is met.

Configuring Delay-Sensitive Packet Interleaving

When you configure CRTP, the software automatically enables link fragmentation and interleaving (LFI). LFI reduces excessive delays by fragmenting long packets into smaller packets and interleaving them with real-time frames. This allows real-time and non-real-time data frames to be carried together on lower-speed links without causing excessive delays to the real-time traffic. When the peer interface receives the smaller fragments, it reassembles the fragments into their original packet. For example, short delay-sensitive packets, such as packetized voice, can race ahead of larger delay-insensitive packets, such as common data packets.

By default, LFI is always active when you include the `compression rtp` statement at the `[edit interfaces` *`interface-name`* `unit` *`logical-unit-number`*`]` hierarchy level. You control the operation of LFI indirectly by setting the `fragment-threshold` statement on the same logical interface. For example, if you include the `fragment-threshold 256` statement at the `[edit interfaces` *`interface-name`* `unit` *`logical-unit-number`*`]` hierarchy level, all IP packets larger than 256 bytes are fragmented.

Example: Configuring Compression of Voice Traffic

Configure compression on a T1 interface with MLPPP encapsulation. Configure fragmentation for all IP packets larger than 128 bytes.

```
[edit interfaces]
t1-1/0/0 {
    unit 0 {
        family mlppp {
            bundle lsq-1/1/0.1;
        }
    }
}
lsq-1/1/0 {
    encapsulation mlppp;
    unit 1 {
        compression {
            rtp {
                port minimum 2000 maximum 64009;
            }
        }
        family inet {
            address 30.1.1.2/24;
        }
        fragment-threshold 128;
    }
}
```

Configuring Encapsulation for Voice Services

Voice services interfaces support the following logical interface encapsulation types:

- Multilink Point-to-Point Protocol (MLPPP), which is the default encapsulation
- ATM2 IQ MLPPP over AAL5 LLC
- Frame Relay PPP

For general information on encapsulation, see the Junos OS Network Interfaces Library for Routing Devices. You can also configure physical interface encapsulation on voice services interfaces.

To configure voice services encapsulation, include the `encapsulation` statement:

```
encapsulation type;
```

You can include this statement at the following hierarchy levels:

- `[edit interfaces interface-name unit logical-unit-number]`
- `[edit logical-systems logical-system-name interfaces interface-name unit logical-unit-number]`

For voice services interfaces, the valid values for the *type* variable are `atm-mlppp-llc`, `frame-relay-ppp` or `multilink-ppp`.

You must also configure the physical interface with the corresponding encapsulation type, either Frame Relay or PPP. LSQ interfaces are supported by the following physical interface types: ATM2 IQ, DS3, E1, E3, OC3, OC12, STM1, and T1, including the channelized versions of these interfaces. For examples, see "Examples: Configuring Voice Services" on page 1052 .

NOTE: The only protocol type supported with `frame-relay-ppp` encapsulation is `family mlppp`.

Configuring Network Interfaces for Voice Services

IN THIS SECTION

- Configuring Voice Services Bundles with MLPPP Encapsulation | **1051**
- Configuring the Compression Interface with PPP Encapsulation | **1052**

To complete a voice services interface configuration, you need to configure the physical network interface with either MLPPP encapsulation and a voice services bundle or PPP encapsulation and a compression interface, as described in the following sections:

Configuring Voice Services Bundles with MLPPP Encapsulation

For voice services interfaces, you configure the link bundle as a channel. The physical interface is usually connected to networks capable of supporting MLPPP; the interface types supported for voice traffic are T1, E1, T3, E3, OC3, OC12, and STM1, including channelized versions of these interfaces.

NOTE: For M Series routers and T Series routers, the following caveats apply:

- Maximum supported throughput on the bundle interfaces is 45 Mbps.
- Bundling of the logical interfaces under a T3 physical interface into the same or different bundles is not supported.

To configure a physical interface link for MLPPP, include the following statement:

```
bundle interface-name;
```

You can configure this statement at the following hierarchy levels:

- [edit interfaces *interface-name* unit *logical-unit-number* family mlppp]
- [edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family mlppp]

When you configure family mlppp, no other protocol configuration is allowed. For more information on link bundles, see *Configuring the Links in a Multilink or Link Services Bundle*.

Configuring the Compression Interface with PPP Encapsulation

To configure the physical interface for PPP encapsulation, you also need to specify the services interface to be used for voice compression: a Link Services IQ (lsq-) interface.

To configure the compression interface, include the compression-device statement:

```
compression-device interface-name;
```

You can configure this statement at the following hierarchy levels:

- [edit interfaces (lsq | ls)-*fpc*/*pic*/*port* unit *logical-unit-number*]
- [edit logical-systems *logical-system-name* interfaces (lsq | ls)-*fpc*/*pic*/*port* unit *logical-unit-number*]

Examples: Configuring Voice Services

Configure voice services using a T1 physical interface and MLPPP bundle encapsulation:

```
[edit interfaces]
t1-0/2/0:1 {
    encapsulation ppp;
    unit 0 {
        family mlppp {
            bundle lsq-1/3/0.1;
        }
    }
}
lsq-1/3/0 {
    unit 1 {
        encapsulation mlppp;
        family inet {
            address 10.5.5.2/30;
        }
        compression {
            rtp {
                f-max-period 100;
                queues [ q1 q2 ];
                port {
                    minimum 16384;
                    maximum 32767;
                }
            }
        }
        fragment-threshold 128;
    }
}
```

Configure voice services using Frame Relay encapsulation without bundling:

```
[edit interfaces]
t1-1/0/0 {
    encapsulation frame-relay;
    unit 0 {
        dlci 100;
        encapsulation frame-relay-ppp;
        compression-device lsq-2/0/0.0;
    }
}
lsq-2/0/0 {
    unit 0 {
        compression {
            rtp {
                f-max-period 100;
```

```
                  queues [ q1 q2 ];
                  port {
                      minimum 16000;
                      maximum 32000;
                  }
              }
          }
          family inet {
              address 10.1.1.1/32;
          }
      }
  }
```

Configure voice services using an ATM2 physical interface (the corresponding class-of-service configuration is provided for illustration):

```
[edit interfaces]
at-1/2/0 {
    atm-options {
        vpi 0;
        pic-type atm2; # only ATM2 PICs are supported
    }
    unit 0 {
        vci 0.69;
        encapsulation atm-mlppp-llc;
        family mlppp {
            bundle lsq-1/3/0.10;
        }
    }
    unit 1 {
        vci 0.42;
        encapsulation atm-mlppp-llc;
        family mlppp {
            bundle lsq-1/3/0.11;
        }
    }
}
lsq-1/3/0 {
    unit 10 {
        encapsulation multilink-ppp;
    }
    # Large packets need to be fragmented.
    # Fragmentation can also be specified per forwarding class.
    fragment-threshold 320;
    compression {
        rtp {
            port minimum 2000 maximum 64009;
        }
    }
}
unit 11 {
    encapsulation multilink-ppp;
}
fragment-threshold 160;
[edit class-of-service]
scheduler-maps {
    sched {
        # Scheduling parameters apply to bundles on the AS or Multiservices PIC.
        # Unlike DS3/SONET interfaces, there is no need to create
        # a separate scheduler map for the ATM PIC. ATM defines
        # CoS constructs under the [edit interfaces at-fpc/pic/port] hierarchy.
        ...
    }
}
fragmentation-maps {
    fragmap {
        forwarding-class {
            ef {
                # In this example, voice is carried in the ef queue.
                # It is interleaved with bulk data.
                # Alternatively, you could use multiclass MLPPP to
                # carry multiple classes of traffic in different
                # multilink classes.
                no-fragmentation;
            }
        }
    }
}
interfaces {
    # Assign fragmentation and scheduling parameters to LSQ interfaces.
    lsq-1/3/0 {
        unit 0 {
            shaping-rate 512k;
            scheduler-map sched;
            fragmentation-map fragmap;
        }
        unit 1 {
            shaping-rate 128k;
            scheduler-map sched;
            fragmentation-map fragmap;
        }
    }
}
```

14

PART

Layer 2 PPP Tunnels

Layer 2 Tunneling of PPP Packets | 1058

CHAPTER 42

Layer 2 Tunneling of PPP Packets

IN THIS CHAPTER

- Layer 2 Tunneling of PPP Packets | **1058**

Layer 2 Tunneling of PPP Packets

IN THIS SECTION

- Layer 2 Tunneling Protocol Overview | **1058**
- L2TP Services Configuration Overview | **1059**
- L2TP Minimum Configuration | **1060**
- Configuring L2TP Tunnel Groups | **1063**
- Configuring the Identifier for Logical Interfaces that Provide L2TP Services | **1068**
- AS PIC Redundancy for L2TP Services | **1070**
- Examples: Configuring L2TP Services | **1071**
- Tracing L2TP Operations | **1075**

Layer 2 Tunneling Protocol Overview

L2TP is defined in RFC 2661, *Layer Two Tunneling Protocol (L2TP).*

L2TP facilitates the tunneling of PPP packets across an intervening network in a way that is as transparent as possible to both end users and applications. It employs access profiles for group and individual user access, and uses authentication to establish secure connections between the two ends of each tunnel. Multilink PPP functionality is also supported.

The L2TP services are supported on the following routers only:

- M7i routers with AS PICs

- M10i routers with AS and MultiServices 100 PICs
- M120 routers with AS, MultiServices 100, and MultiServices 400 PICs
- On MX Series routers, the L2TP access concentrator (LAC) and L2TP network server (LNS) functions are supported only on MPCs; they are not supported on any services PIC or MS-DPC. For details about MPC support for L2TP, see the *MX Series Interface Module Reference*

For more information, see "L2TP Services Configuration Overview" on page 1059 .

SEE ALSO

AS PIC Redundancy for L2TP Services | **1070**

L2TP Services Configuration Overview

The statements for configuring L2TP services are found at the following hierarchy levels:

- [edit services l2tp tunnel-group *group-name*]

  The L2TP **tunnel-group** statement identifies an L2TP instance or L2TP server. Associated statements specify the local gateway address on which incoming tunnels and sessions are accepted, the Adaptive Services (AS) *Physical Interface Card* (PIC) that processes data for the sessions in this tunnel group, references to L2TP and PPP access profiles, and other attributes for configuring window sizes and timer values.

- [edit interfaces sp-*fpc*/*pic*/*port* unit *logical-unit-number* dial-options]

  The **dial-options** statement includes configuration for the **l2tp-interface-id** statement and the **shared**/**dedicated** flag. The interface identifier associates a user session with a *logical interface*. Sessions can use either shared or dedicated logical interfaces. To run routing protocols, a session must use a dedicated logical interface.

- [edit access profile *profile-name* client *name* l2tp]

  Tunnel profiles are defined at the **[edit access]** hierarchy level. Tunnel clients are defined with authentication, multilink negotiation and fragmentation, and other L2TP attributes in these profiles.

- [edit access profile *profile-name* client *name* ppp]

  User profiles are defined at the **[edit access]** hierarchy level. User clients are defined with authentication and other PPP attributes in these profiles. These client profiles are used when local authentication is specified.

- [edit access radius-server *address*]

When you configure **authentication-order radius** at the **[edit access profile *profile-name*]** hierarchy level, you must configure a RADIUS service at the **[edit access radius-server]** hierarchy level.

NOTE: For more information about configuring properties at the **[edit access]** hierarchy level, see the Junos OS Administration Library for Routing Devices. For information about L2TP LAC and LNS configurations on MX Series routers for subscriber access, see *L2TP for Subscriber Access Overview* in the *Junos Subscriber Access Configuration Guide*.

L2TP Minimum Configuration

To configure L2TP services, you must perform at least the following tasks:

- Define a tunnel group at the `[edit services l2tp]` hierarchy level with the following attributes:
    - `l2tp-access-profile`—Profile name for the L2TP tunnel.
    - `ppp-access-profile`—Profile name for the L2TP user.
    - `local-gateway`—Address for the L2TP tunnel.
    - `service-interface`—AS PIC interface for the L2TP service.
    - Optionally, you can configure `traceoptions` for debugging purposes.

  The following example shows a minimum configuration for a tunnel group with trace options:

```
[edit services l2tp]
tunnel-group finance-lns-server {
    l2tp-access-profile westcoast_bldg_1_tunnel;
    ppp-access-profile westcoast_bldg_1;
    local-gateway {
        address 10.21.255.129;
    }
    service-interface sp-1/3/0;
}
traceoptions {
    flag all;
    filter {
        protocol udp;
        protocol l2tp;
        protocol ppp;
        protocol radius;
    }
}
```

- At the [edit interfaces] hierarchy level:
  - Identify the physical interface at which L2TP tunnel packets enter the router, for example ge-0/3/0.
  - Configure the AS PIC interface with unit 0 family inet defined for IP service, and configure another logical interface with family inet and the dial-options statement.

  The following example shows a minimum interfaces configuration for L2TP:

```
[edit interfaces]
ge-0/3/0 {
    unit 0 {
        family inet {
            address 10.58.255.129/28;
        }
    }
}
sp-1/3/0 {
    unit 0 {
        family inet;
    }
    unit 20 {
        dial-options {
            l2tp-interface-id test;
            shared;
        }
        family inet;
    }
}
```

- At the [edit access] hierarchy level:
  - Configure a tunnel profile. Each client specifies a unique L2TP Access Concentrator (LAC) name with an interface-id value that matches the one configured on the AS PIC interface unit; shared-secret is authentication between the LAC and the L2TP Network Server (LNS).
  - Configure a user profile. If RADIUS is used as the authentication method, it needs to be defined.
  - Define the RADIUS server with an IP address, port, and authentication data shared between the router and the RADIUS server.

NOTE: When the L2TP Network Server (LNS) is configured with RADIUS authentication, the default behavior is to accept the preferred RADIUS-assigned IP address. Previously, the default behavior was to accept and install the nonzero peer IP address that came into the IP-Address option of the IPCP Configuration Request packet.

- Optionally, you can define a group profile for common attributes, for example `keepalive 0` to turn off keepalive messages.

The following example shows a minimum profiles configuration for L2TP:

```
[edit access]
group-profile westcoast_users {
    ppp {
        keepalive 0;
    }
}
profile westcoast_bldg_1_tunnel {
    client production {
        l2tp {
            interface-id test;
            shared-secret "$ABC123"; # SECRET-DATA
        }
        user-group-profile westcoast_users;
    }
}
profile westcoast_bldg_1 {
    authentication-order radius;
}
radius-server {
    192.168.65.63 {
        port 1812;
        secret "$ABC123"; # SECRET-DATA
    }
}
```

SEE ALSO

Configuring the Identifier for Logical Interfaces that Provide L2TP Services | **1068**

Tracing L2TP Operations | **1075**

Configuring L2TP Tunnel Groups

IN THIS SECTION

- Configuring Access Profiles for L2TP Tunnel Groups | **1063**
- Configuring the Local Gateway Address and PIC | **1064**
- Configuring Window Size for L2TP Tunnels | **1064**
- Configuring Timers for L2TP Tunnels | **1065**
- Hiding Attribute-Value Pairs for L2TP Tunnels | **1066**
- Configuring System Logging of L2TP Tunnel Activity | **1066**

To establish L2TP service on a router, you need to identify an L2TP tunnel group and specify a number of values that define which access profiles, interface addresses, and other properties to use in creating a tunnel. To identify the tunnel group, include the `tunnel-group` statement at the `[edit services l2tp]` hierarchy level.

NOTE: If you delete a tunnel group or mark it inactive, all L2TP sessions in that tunnel group are terminated. If you change the value of the `local-gateway address` or the `service-interface` statement, all L2TP sessions using those settings are terminated. If you change or delete other statements at the `[edit services l2tp tunnel-group` *`group-name`*`]` hierarchy level, new tunnels you establish will use the updated values but existing tunnels and sessions are not affected.

This following sections explain how to configure L2TP tunnel groups:

Configuring Access Profiles for L2TP Tunnel Groups

To validate L2TP connections and session requests, you set up access profiles by configuring the `profile` statement at the `[edit access]` hierarchy level. You need to configure two types of profiles:

- L2TP tunnel access profile, which validates all L2TP connection requests to the specified local gateway address
- PPP access profile, which validates all PPP session requests through L2TP tunnels established to the local gateway address

For more information on configuring the profiles, see the Junos OS Administration Library for Routing Devices. A profile example is included in "Examples: Configuring L2TP Services" on page 1071 .

To associate the profiles with a tunnel group, include the `l2tp-access-profile` and `ppp-access-profile` statements at the `[edit services l2tp tunnel-group group-name]` hierarchy level:

```
l2tp-access-profile profile-name;
ppp-access-profile profile-name;
```

Configuring the Local Gateway Address and PIC

When you configure an L2TP group, you must also define a local address for the L2TP tunnel connections and the AS PIC that processes the requests:

- To configure the local gateway IP address, include the `address` statement at the `[edit services l2tp tunnel-group group-name local-gateway]` hierarchy level:

  ```
  address address;
  ```

- To configure the AS PIC, include the `service-interface` statement at the `[edit services l2tp tunnel-group group-name]` hierarchy level:

  ```
  service-interface sp-fpc/pic/port;
  ```

You can optionally specify the logical unit number along with the service interface. If specified, the unit is used as a logical interface representing PPP sessions negotiated using this profile.

NOTE: If you change the local gateway address or the service interface configuration, all L2TP sessions using those settings are terminated.

Dynamic class-of-service (CoS) functionality is supported on L2TP LNS sessions or L2TP sessions with ATM VCs, as long as the L2TP session is configured to use an IQ2 PIC on the egress interface. For more information, see the Class of Service User Guide (Routers and EX9200 Switches).

Configuring Window Size for L2TP Tunnels

You can configure the maximum window size for packet processing at each end of the L2TP tunnel:

- The receive window size limits the number of concurrent packets the server processes. By default, the maximum is 16 packets. To change the window size, include the `receive-window` statement at the `[edit services l2tp tunnel-group group-name]` hierarchy level:

  ```
  receive-window packets;
  ```

- The maximum-send window size limits the other end's receive window size. The information is transmitted in the receive window size attribute-value pair. By default, the maximum is 32 packets. To change the window size, include the `maximum-send-window` statement at the `[edit services l2tp tunnel-group group-name]` hierarchy level:

  ```
  maximum-send-window packets;
  ```

Configuring Timers for L2TP Tunnels

You can configure the following timer values that regulate L2TP tunnel processing:

- Hello interval—If the server does not receive any messages within a specified time interval, the router software sends a hello message to the tunnel's remote peer. By default, the interval length is 60 seconds. If you configure a value of 0, no hello messages are sent. To configure a different value, include the `hello-interval` statement at the `[edit services l2tp tunnel-group group-name]` hierarchy level:

  ```
  hello-interval seconds;
  ```

- Retransmit interval—By default, the retransmit interval length is 30 seconds. To configure a different value, include the `retransmit-interval` statement at the `[edit services l2tp tunnel-group group-name]` hierarchy level:

  ```
  retransmit-interval seconds;
  ```

- Tunnel timeout—If the server cannot send any data through the tunnel within a specified time interval, it assumes that the connection with the remote peer has been lost and deletes the tunnel. By default, the interval length is 120 seconds. To configure a different value, include the `tunnel-timeout` statement at the `[edit services l2tp tunnel-group group-name]` hierarchy level:

  ```
  tunnel-timeout seconds;
  ```

Hiding Attribute-Value Pairs for L2TP Tunnels

Once an L2TP tunnel has been established and the connection authenticated, information is encoded by means of attribute-value pairs. By default, this information is not hidden. To hide the attribute-value pairs once the shared secret is known, include the `hide-avps` statement at the `[edit services l2tp tunnel-group group-name]` hierarchy level:

```
hide-avps;
```

Configuring System Logging of L2TP Tunnel Activity

You can specify properties that control how system log messages are generated for L2TP services.

To configure interface-wide default system logging values, include the `syslog` statement at the `[edit services l2tp tunnel-group group-name]` hierarchy level:

```
syslog {
    host hostname {
        services severity-level;
        facility-override facility-name;
        log-prefix prefix-value;
    }
}
```

Configure the `host` statement with a hostname or IP address that specifies the system log target server. The hostname `local` directs system log messages to the Routing Engine. For external system log servers, the hostname must be reachable from the same routing instance to which the initial data packet (that triggered session establishment) is delivered. You can specify only one system logging hostname.

Table 32 on page 1066 lists the severity levels that you can specify in configuration statements at the `[edit services l2tp tunnel-group group-name syslog host hostname]` hierarchy level. The levels from `emergency` through `info` are in order from highest severity (greatest effect on functioning) to lowest.

Table 32: System Log Message Severity Levels

Severity Level	Description
any	Includes all severity levels

Table 32: System Log Message Severity Levels *(Continued)*

Severity Level	Description
`emergency`	System panic or other condition that causes the router to stop functioning
`alert`	Conditions that require immediate correction, such as a corrupted system database
`critical`	Critical conditions, such as hard drive errors
`error`	Error conditions that generally have less serious consequences than errors in the emergency, alert, and critical levels
`warning`	Conditions that warrant monitoring
`notice`	Conditions that are not errors but might warrant special handling
`info`	Events or nonerror conditions of interest

We recommend setting the system logging severity level to `error` during normal operation. To monitor PIC resource usage, set the level to `warning`. To gather information about an intrusion attack when an intrusion detection system error is detected, set the level to `notice` for a specific service set. To debug a configuration or log Network Address Translation (NAT) events, set the level to `info`.

For more information about system log messages, see the System Log Explorer.

To use one particular facility code for all logging to the specified system log host, include the `facility-override` statement at the `[edit services l2tp tunnel-group` *`group-name`* `syslog host` *`hostname`*`]` hierarchy level:

```
facility-override facility-name;
```

The supported facilities include: `authorization`, `daemon`, `ftp`, `kernel`, `user`, and `local0` through `local7`.

To specify a text prefix for all logging to this system log host, include the `log-prefix` statement at the `[edit services l2tp tunnel-group` *`group-name`* `syslog host` *`hostname`*`]` hierarchy level:

```
log-prefix prefix-text;
```

Configuring the Identifier for Logical Interfaces that Provide L2TP Services

IN THIS SECTION

- Example: Configuring Multilink PPP on a Shared Logical Interface | **1069**

You can configure L2TP services on adaptive services interfaces on M7i, M10i, M120, and MX Series routers only. You must configure the logical interface to be dedicated or shared. If a logical interface is dedicated, it can represent only one session at a time. A shared logical interface can have multiple sessions.

To configure the logical interface, include the `l2tp-interface-id` statement at the `[edit interfaces interface-name unit logical-unit-number dial-options]` hierarchy level:

```
l2tp-interface-id name;
(dedicated | shared);
```

The `l2tp-interface-id` name configured on the logical interface must be replicated at the `[edit access profile name]` hierarchy level:

- For a user-specific identifier, include the `l2tp-interface-id` statement at the `[edit access profile name ppp]` hierarchy level.
- For a group identifier, include the `l2tp-interface-id` statement at the `[edit access profile name l2tp]` hierarchy level.

You can configure multiple logical interfaces with the same interface identifier, to be used as a pool for several users. For more information on configuring access profiles, see the Junos OS Administration Library for Routing Devices.

NOTE: If you delete the `dial-options` statement settings configured on a logical interface, all L2TP sessions running on that interface are terminated.

Example: Configuring Multilink PPP on a Shared Logical Interface

Multilink PPP is supported on either shared or dedicated logical interfaces. The following example can be used to configure many multilink bundles on a single shared interface:

```
interfaces {
    sp-1/3/0 {
        traceoptions {
            flag all;
        }
        unit 0 {
            family inet;
        }
        unit 20 {
            dial-options {
                l2tp-interface-id test;
                shared;
            }
            family inet;
        }
    }
}
access {
    profile t {
        client test {
            l2tp {
                interface-id test;
                multilink;
                shared-secret "$ABC123"; # SECRET-DATA
            }
        }
    }
    profile u {
        authentication-order radius;
    }
    radius-server {
        192.168.65.63 {
            port 1812;
            secret "$ABC123"; # SECRET-DATA
        }
    }
}
```

```
services {
    l2tp {
        tunnel-group 1 {
            l2tp-access-profile t;
            ppp-access-profile u;
            local-gateway {
                address 10.70.1.1;
            }
            service-interface sp-1/3/0;
        }
        traceoptions {
            flag all;
            debug-level packet-dump;
            filter {
                protocol l2tp;
                protocol ppp;
                protocol radius;
            }
        }
    }
}
```

AS PIC Redundancy for L2TP Services

L2TP services support AS PIC redundancy. To configure redundancy, you specify a redundancy services PIC (`rsp`) interface in which the primary AS PIC is active and a secondary AS PIC is on standby. If the primary AS PIC fails, the secondary PIC becomes active, and all service processing is transferred to it. If the primary AS PIC is restored, it remains in standby and does not preempt the secondary AS PIC; you need to manually restore the services to the primary PIC. To determine which PIC is currently active, issue the `show interfaces redundancy` command.

NOTE: On L2TP, the only service option supported is *warm standby*, in which one backup PIC supports multiple working PICs. Recovery times are not guaranteed, because the configuration must be completely restored on the backup PIC after a failure is detected. The tunnels and sessions are torn down upon switchover and need to be restarted by the LAC and PPP client, respectively. However, configuration is preserved and available on the new active PIC, although the protocol state needs to be reestablished.

As with the other AS PIC services that support warm standby, you can issue the `request interfaces (revert | switchover)` command to manually switch between primary and secondary L2TP interfaces.

For more information, see "Configuring AS or Multiservices PIC Redundancy" on page 1042 . For an example configuration, see "Examples: Configuring L2TP Services" on page 1071 . For information on operational mode commands, see the CLI Explorer.

Examples: Configuring L2TP Services

Configure L2TP with multiple group and user profiles and a pool of logical interfaces for concurrent tunnel sessions:

```
[edit access]
address-pool customer_a {
    address 10.1.1.1/32;
}
address-pool customer_b {
    address-range low 10.2.2.1 high 10.2.3.2;
}
group-profile sunnyvale_users {
    ppp {
        framed-pool customer_a;
        idle-timeout 15;
        primary-dns 192.168.65.1;
        secondary-dns 192.168.65.2;
        primary-wins 192.168.65.3;
        secondary-wins 192.168.65.4;
        interface-id west;
    }
}
group-profile eastcoast_users {
    ppp {
        framed-pool customer_b;
        idle-timeout 20;
        primary-dns 192.168.65.5;
        secondary-dns 192.168.65.6;
        primary-wins 192.168.65.7;
        secondary-wins 192.168.65.8;
        interface-id east;
    }
}
group-profile sunnyvale_tunnel {
    l2tp {
        maximum-sessions-per-tunnel 100;
        interface-id west_shared;
    }
}
group-profile east_tunnel {
    l2tp {
        maximum-sessions-per-tunnel 125;
        interface-id east_shared;
    }
}
profile sunnyvale_bldg_1 {
    client white {
        chap-secret "$ABC123"; #            SECRET-DATA
        ppp {
            idle-timeout 22;
            primary-dns 192.168.65.1;
            framed-ip-address 10.12.12.12/32;
            interface-id east;
        }
        group-profile sunnyvale_users;
    }
    client blue {
        chap-secret "$ABC123"; #           SECRET-DATA
        group-profile sunnyvale_users;
    }
    authentication-order password;
}
profile sunnyvale_bldg_1_tunnel {
    client test {
        l2tp {
            shared-secret "$ABC123"; # SECRET-DATA
            maximum-sessions-per-tunnel 75;
            interface-id west_shared;
            ppp-authentication chap;
        }
        group-profile sunnyvale_tunnel;
    }
    client production {
        l2tp {
            shared-secret "$ABC123";
            ppp-authentication chap;
        }
        group-profile sunnyvale_tunnel;
    }
}
[edit services]
```

```
l2tp {
    tunnel-group finance-lns-server {
        l2tp-access-profile sunnyvale_bldg_1_tunnel;
        ppp-access-profile sunnyvale_bldg_1;
        local-gateway {
            address 10.1.117.3;
        }
        service-interface sp-1/3/0;
        receive-window 1500;
        maximum-send-window 1200;
        retransmit-interval 5;
        hello-interval 15;
        tunnel-timeout 55;
    }
    traceoptions {
        flag all;
    }
}
[edit interfaces sp-1/3/0]
unit0 {
    family inet;
}
unit 10 {
    dial-options {
        l2tp-interface-id foo-user;
        dedicated;
    }
    family inet;
}
unit 11 {
    dial-options {
        l2tp-interface-id east;
        dedicated;
    }
    family inet;
}
unit 12 {
    dial-options {
        l2tp-interface-id east;
        dedicated;
    }
    family inet;
}
```

```
unit 21 {
    dial-options {
        l2tp-interface-id west;
        dedicated;
    }
    family inet;
}
unit 30 {
    dial-options {
        l2tp-interface-id west_shared;
        shared;
    }
    family inet;
}
unit 40 {
    dial-options {
        l2tp-interface-id east_shared;
        shared;
    }
    family inet;
}
```

Configure L2TP redundancy:

```
interfaces {
    rsp0 {
        redundancy-options {
            primary sp-0/0/0;
            secondary sp-1/3/0;
        }
        unit 0 {
            family inet;
        }
        unit 11 {
            dial-options {
                l2tp-interface-id east_shared;
                shared;
            }
            family inet;
        }
    }
}
```

Tracing L2TP Operations

Tracing operations track all AS PIC operations and record them in a log file in the **/var/log** directory. By default, this file is named **/var/log/l2tpd**.

> **NOTE**: This topic refers to tracing L2TP LNS operations on M Series routers. To trace L2TP LAC operations on MX Series routers, see *Tracing L2TP Events for Troubleshooting*.

To trace L2TP operations, include the `traceoptions` statement at the `[edit services l2tp]` hierarchy level:

```
traceoptions {
    debug-level level;
    file <filename> <files number> <match regular-expression > <size maximum-file-size> <world-
readable | no-world-readable>;
    filter {
        protocol name;
        user-name username;
    }
    flag flag;
    interfaces interface-name {
        debug-level severity;
        flag flag;
    }
    level (all | error | info | notice | verbose | warning);
    no-remote-trace;
}
```

You can specify the following L2TP tracing flags:

- `all`—Trace everything.
- `configuration`—Trace configuration events.
- `protocol`—Trace routing protocol events.
- `routing-socket`—Trace routing socket events.
- `rpd`—Trace routing protocol process events.

You can specify a trace level for PPP, L2TP, RADIUS, and User Datagram Protocol (UDP) tracing. To configure a trace level, include the `debug-level` statement at the `[edit services l2tp traceoptions]` hierarchy level and specify one of the following values:

- `detail`—Detailed debug information
- `error`—Errors only
- `packet-dump`—Packet decoding information

You can filter by protocol. To configure filters, include the `filter protocol` statement at the `[edit services l2tp traceoptions]` hierarchy level and specify one or more of the following protocol values:

- `ppp`
- `l2tp`
- `radius`
- `udp`

To implement filtering by protocol name, you must also configure either `flag protocol` or `flag all`.

You can also configure traceoptions for L2TP on a specific adaptive services interface. To configure per-interface tracing, include the `interfaces` statement at the `[edit services l2tp traceoptions]` hierarchy level:

```
interfaces interface-name {
    debug-level level;
    flag flag;
}
```

NOTE: Implementing traceoptions consumes CPU resources and affects the packet processing performance.

You can specify the `debug-level` and `flag` statements for the interface, but the options are slightly different from the general L2TP traceoptions. You specify the debug level as `detail`, `error`, or `extensive`, which provides complete PIC debug information. The following flags are available:

- `all`—Trace everything.
- `ipc`—Trace L2TP Inter-Process Communication (IPC) messages between the PIC and the Routing Engine.
- `packet-dump`—Dump each packet's content based on debug level.

- `protocol`—Trace L2TP, PPP, and multilink handling.
- `system`—Trace packet processing on the PIC.

15

PART

URL Filtering

URL Filtering | 1079

CHAPTER 43

URL Filtering

IN THIS CHAPTER

- URL Filtering | 1079

URL Filtering

IN THIS SECTION

- URL Filtering Overview | 1079
- Configuring URL Filtering | 1085
- DNS Request Filtering for Disallowed Website Domains | 1090
- Integration of Juniper ATP Cloud and Web Filtering on MX Series Routers | 1111

URL Filtering Overview

IN THIS SECTION

- URL Filter Database File | 1082
- URL Filter Profile Caveats | 1083

You can use URL filtering to determine which Web content is not accessible to users.

Components of this feature include the following:

- URL filter database file

- Configuration of one or more templates (up to eight per profile)
- URL Filter Plug-in (jservices-urlf)
- URL filtering daemon (url-filterd)

The URL filter database file is stored on the Routing Engine and contains all the disallowed URLs. Configured *templates* define which traffic to monitor, what criteria to match, and which actions to take. You configure the templates and the location of the URL filter database file in a *profile*.

Starting in Junos OS Release 17.2R2 and 17.4R1, for Adaptive Services, you can disable the filtering of HTTP traffic that contains an embedded IP address (for example, http:/10.1.1.1) belonging to a disallowed domain name in the URL filter database.Starting in Junos OS Release 19.3R2, this same functionaly is supported for Next Gen Services on MX240, MX480, and MX960.

To enable the URL filtering feature, you must configure `jservices-urlf` as the *package-name* at the `[edit chassis fpc` *slot-number* `pic` *pic-number* `adaptive-services service-package extension-provider]` hierarchy level. Once enabled, jservices-urlf maintains the URL filtering profile and receives all traffic to be filtered, the filtering criteria, and the action to be taken on the filtered traffic.

NOTE: MX-SPC3 does not explicitly need `jservices-urlf` as the *package-name* at the `[edit chassis fpc` *slot-number* `pic` *pic-number* `adaptive-services service-package extension-provider]` hierarchy level. It is supported by default.

The URL filtering daemon (url-filterd), which also resides on the Routing Engine, resolves the domain name of each URL in the URL filter database to a list of IPv4 and IPv6 addresses. It then downloads the list of IP addresses to the service PIC, which runs jservices-urlf. Then url-filterd interacts with the Dynamic Firewall process (dfwd) to install filters on the Packet Forwarding Engine to punt the selected traffic from the Packet Forwarding Engine to the service PIC.

As new HTTP and HTTPS traffic reaches the router, a decision is made based on the information in the URL filter database file. The filtering rules are checked and either the router accepts the traffic and passes it on or blocks the traffic. If the traffic is blocked, one of the following configured actions is taken:

- An HTTP redirect is sent to the user.
- A custom page is sent to the user.
- An HTTP status code is sent to the user.
- A TCP reset is sent.

Accept is also an option. In this case, the traffic is not blocked.

Figure 48 on page 1081 illustrates the URL filtering for HTTP sessions.

Figure 48: Packet Flow-URL Filtering for HTTP Sessions

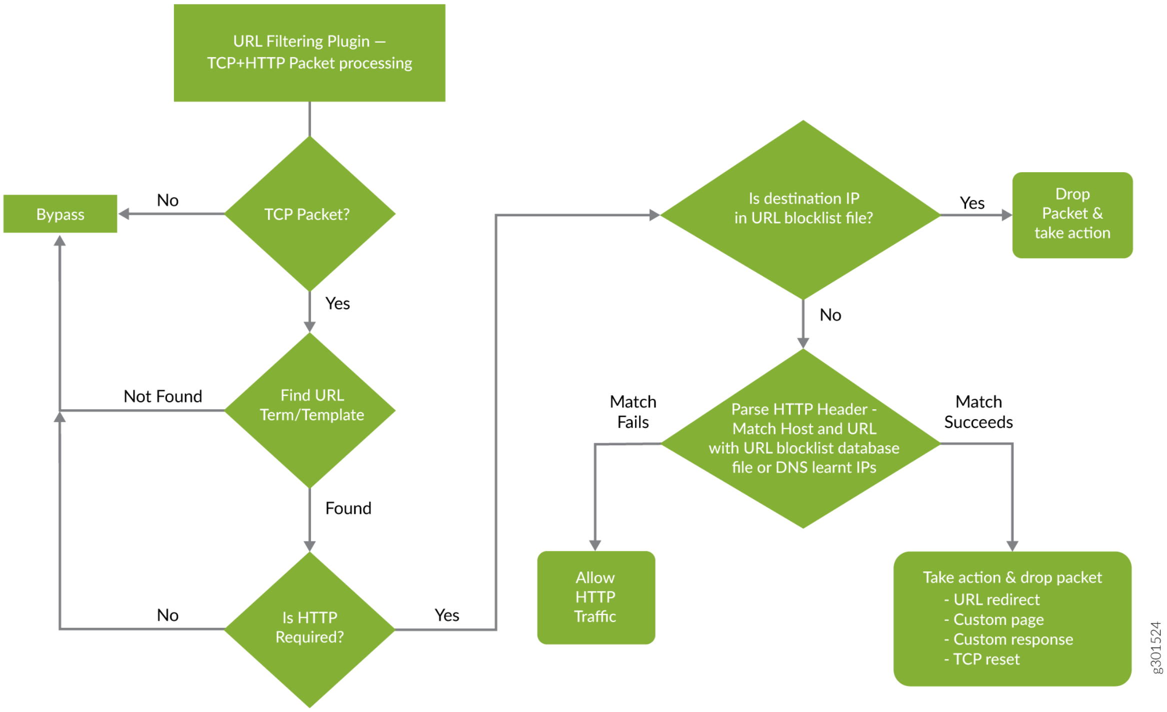
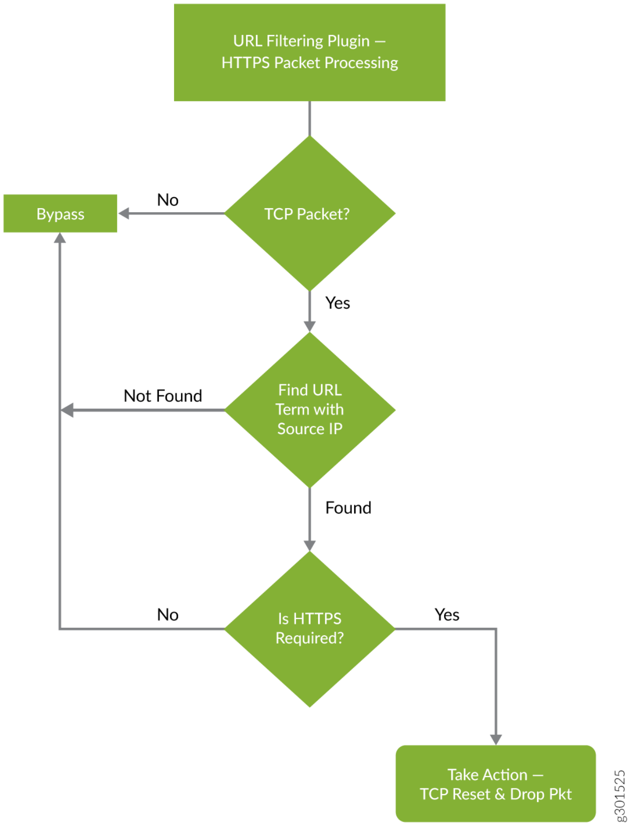


Figure 49 on page 1082 illustrates the URL filtering for HTTPS sessions.

Figure 49: Packet Flow-URL Filtering for HTTPS Sessions



For more details on the URL filtering feature, see the following sections:

URL Filter Database File

The URL filter database file contains entries of URLs and IP addresses. Create the URL filter database file in the format indicated in Table 33 on page 1082 and locate it on the Routing Engine in the **/var/db/url-filterd** directory.

Table 33: URL Filter Database File Format

Entry	Description	Example
FQDN	Fully qualified domain name.	www.badword.com/jjj/bad.jpg

Table 33: URL Filter Database File Format *(Continued)*

Entry	Description	Example
URL	Full string URL without the Layer 7 protocol.	www.srch.com/*badword*/ www.srch.com www.srch.com/xyz www.srch.com/xyz*
IPv4 address	HTTP request on a specific IPv4 address.	10.1.1.199
IPv6 address	HTTP request on a specific IPv6 address.	1::1

You must specify a custom URL filter database in the profile. If needed, you can also assign a custom URL filter database file with any template, and that database takes precedence over the database configured at the profile level.

If you change the contents of the URL filter database file, use the `request services (url-filter | web-filter) update` command. Other commands to help maintain the URL filter database file include the following:

- `request services (url-filter | web-filter) delete`
- `request services (url-filter | web-filter) force`
- `request services (url-filter | web-filter) validate`

URL Filter Profile Caveats

The URL filter profile consists of from one to eight templates. Each template consists of a set of configured logical interfaces where traffic is monitored for URL filtering and one or more terms.

A *term* is a set of match criteria with actions to be taken if the match criteria is met. You must configure at least one term to configure URL filtering. Each term consists of a `from` statement and a `then` statement, where the `from` statement defines the source IP prefixes and destination ports that are monitored. The `then` statement specifies the action to be taken. If you omit the `from` statement, any source IP prefix and any destination port are considered to match. But you can omit only one `from` statement per template or per profile.

Example configuration of multiple terms without from statements

```
template1 {
    client-interfaces [ xe-4/0/3.35 xe-4/0/3.36 ];
    server-interfaces xe-4/0/0.31;
    dns-source-interface xe-4/0/0.1;
    dns-routing-instance data_vr;
    routing-instance data_vr2;
    dns-server 50.0.0.3;
    dns-retries 3;
    url-filter-database url_database.txt;
    term term1 {
        then {
            tcp-reset;
        }
    }
    term term2 {
        then {
            redirect-url www.google.com;
        }
    }
}
```

If you omit more than one `from` statement per template, you will get the following error message on commit:

```
URLFD_CONFIG_FAILURE: Configuration not valid:
Cannot have two wild card terms in template template1
error: configuration check-out failed
```

SEE ALSO

- *request services url-filter update url-filter-database file*
- *request services url-filter force dns-resolution*
- *request services url-filter delete gencfg-data*
- *request services url-filter validate*

Configuring URL Filtering

To configure the URL filtering feature, you must first configure `jservices-urlf` as the *package-name* at the `[edit chassis fpc slot-number pic pic-number adaptive-services service-package extension-provider]` hierarchy level. For more information on configuring the `extension-provider package package-name` configuration statement, see the *package (Loading on PIC)* statement.

> **NOTE**: MX-SPC3 does not explicitly need `jservices-urlf` as the *package-name* at the `[edit chassis fpc slot-number pic pic-number adaptive-services service-package extension-provider]` hierarchy level. It is supported by default.

URL filtering is configured on a service PIC. The interfaces you are dealing with are services interfaces (which use the `ms` prefix) or aggregated multiservices (AMS) interfaces (which use the `ams` prefix). For more information on AMS interfaces, see the *Adaptive Services Interfaces User Guide for Routing Devices* starting with "Understanding Aggregated Multiservices Interfaces" on page 1010 .

A URL filtering *profile* is a collection of templates. Each template consists of a set of criteria that defines which URLs are disallowed and how the recipient is notified.

To configure the URL profile:

1. Assign a name to the URL profile.

   ```
   [edit]
   user@host# edit services (web-filter | url-filter) profile profile-name
   ```

   Starting in Junos OS Release 18.3R1, for Adaptive Services. configure the profile at the `[edit services web-filter]` hierarchy level. Before Junos OS Release 18.3R1, configure the profile at the `[edit services url-filter]` hierarchy level.Starting in Junos OS Release 19.3R2, this same functionality is available for Next Gen Serices on MX240, MX480, and MX960.

2. Specify the name of the URL filter database to use.

   ```
   [edit services (web-filter | url-filter) profile profile-name]
   user@host# set url-filter-database filename
   ```

3. Configure one or more templates for the profile.

   To configure each template:

a. Name the template.

```
[edit services (web-filter | url-filter) profile profile-name]
user@host# set (url-filter-template template-name | template template-name)
```

NOTE: Starting in Junos OS Release 18.3R1, configure the template with the `url-filter-template` statement. Before Junos OS Release 18.3R1, configure the template with the `template` statement.

b. Go to that new template hierarchy level.

```
[edit services (web-filter | url-filter) profile profile-name]
user@host# edit (url-filter-template template-name | template template-name)
```

c. Specify the name of the URL filter database to use.

```
[edit services (web-filter | url-filter) profile profile-name (url-filter-template
template-name | template template-name)]
user@host# set url-filter-database filename
```

d. Specify the loopback interface for which the source IP address is picked for sending DNS queries.

```
[edit services (web-filter | url-filter) profile profile-name (url-filter-template
template-name | template template-name)]
user@host# set dns-source-interface loopback-interface-name
```

e. Disable the filtering of HTTP traffic that contains an embedded IP address (for example, http:/10.1.1.1) belonging to a disallowed domain name in the URL filter database.

```
[edit services (web-filter | url-filter) profile profile-name (url-filter-template
template-name | template template-name)]
user@host# set disable-url-filtering
```

f. Configure the DNS resolution time interval in minutes.

```
[edit services (web-filter | url-filter) profile profile-name (url-filter-template
template-name | template template-name)]
user@host# set dns-resolution-interval minutes
```

g. Configure the number of retries for a DNS query in case the query fails or times out.

```
[edit services (web-filter | url-filter) profile profile-name]
user@host# set dns-retries number
```

h. Specify the IP addresses (IPv4 or IPv6) of DNS servers to which the DNS queries are sent.

```
[edit services (web-filter | url-filter) profile profile-name (url-filter-template
template-name | template template-name)]
user@host# set dns-server [ip-address]
```

i. Specify the client-facing logical interfaces on which the URL filtering is configured.

```
[edit services (web-filter | url-filter) profile profile-name (url-filter-template
template-name | template template-name)]
user@host# set client-interfaces [ client-interface-name ]
```

j. Specify the server-facing logical interfaces on which the URL filtering is configured.

```
[edit services (web-filter | url-filter) profile profile-name (url-filter-template
template-name | template template-name)]
user@host# set server-interfaces [ server-interface-name ]
```

k. Specify the routing instance on which the URL filtering is configured.

```
[edit services (web-filter | url-filter) profile profile-name (url-filter-template
template-name | template template-name)]
user@host# set routing-instance routing-instance-name
```

l. Specify the routing instance on which the DNS server is reachable.

```
[edit services (web-filter | url-filter) profile profile-name (url-filter-template
template-name | template template-name)]
user@host# dns-routing-instance dns-routing-instance-name
```

4. Configure the term information.

   Terms are used in filters to segment the policy or filter into small match and action pairs.

   a. Name the term.

   ```
   [edit services (web-filter | url-filter) profile profile-name (url-filter-template
   template-name | template template-name)]
   user@host# set term term-name
   ```

   b. Go to the new term hierarchy level.

   ```
   [edit services (web-filter | url-filter) profile profile-name (url-filter-template
   template-name | template template-name)]
   user@host# edit term term-name
   ```

   c. Specify the source IP address prefixes for traffic you want to filter.

   ```
   [edit services (web-filter | url-filter) profile profile-name (url-filter-template
   template-name | template template-name) term term-name]
   user@host# set from src-ip-prefix [prefix]
   ```

   d. Specify the destination ports for traffic you want to filter.

   ```
   [edit services  (web-filter | url-filter) profile profile-name (url-filter-template
   template-name | template template-name) term term-name]
   user@host# set from dest-port [port]
   ```

e. Configure an action to take.

```
[edit services  (web-filter | url-filter) profile profile-name (url-filter-template
template-name | template template-name) term term-name]
user@host# set then action
```

The action can be one of the following:

`custom-page` *`custom-page`*	Send a custom page string to the user.
`http-status-code` *`http-status-code`*	Send an HTTP status code to the user.
`redirect-url` *`redirect-url`*	Send an HTTP redirect to the user.
`tcp-reset`	Send a TCP reset to the user.

5. Associate the URL profile with a next-hop service set.

NOTE: For URL filtering, you must configure the service set as a next-hop service set.

```
[edit]
user@host# set services service-set service-set-name (web-filter-profile profile-name | url-
filter-profile profile-name)
user@host# set services service-set service-set-name next-hop-service inside-service-
interface interface-name.unit-number
user@host# set services service-set service-set-name next-hop-service outside-service-
interface interface-name.unit-number
```

NOTE: The service interface can also be of the `ams` prefix. If you are using `ams` interfaces at the `[edit services service-set` *`service-set-name]`* hierarchy level for the URL filter, you must also configure the `load-balancing-options hash-keys` statement at the `[edit interfaces` *`ams-interface-name`* `unit` *`number`*`]` hierarchy level. .

NOTE: Starting in Junos OS Release 18.3R1, configure the service set with the `web-filter-profile` statement. Before Junos OS Release 18.3R1, configure the service set with the `url-filter-profile` statement.

SEE ALSO

Configuring Service Sets to be Applied to Services Interfaces | 10

DNS Request Filtering for Disallowed Website Domains

IN THIS SECTION

- Overview of DNS Request Filtering | 1090
- How to Configure DNS Request Filtering | 1092
- Multitenant Support for DNS Filtering | 1100
- Configuring Multi-tenant Support for DNS Filtering | 1101
- Example: Configuring Multitenant Support for DNS Filtering | 1106

Overview of DNS Request Filtering

IN THIS SECTION

- Benefits | 1092
- Disallowed Domain Filter Database File | 1092
- DNS Filter Profile | 1092

Starting in Junos OS Release 18.3R1, you can configure DNS filtering to identify DNS requests for disallowed website domains. Starting in Junos OS Release 19.3R2, you can configure DNS filtering if you are running Next Gen Services with the MX-SPC3 services card. Next Gen Services are supported on MX240, MX480 and MX960 routers. For DNS request types A, AAAA, MX, CNAME, TXT, SRV, and ANY, you configure the action to take for a DNS request for a disallowed domain. You can either:

- Block access to the website by sending a DNS response that contains the IP address or fully qualified domain name (FQDN) of a DNS sinkhole server. This ensures that when the client attempts to send traffic to the disallowed domain, the traffic instead goes to the sinkhole server (see Figure 3).

- Log the request and allow access.

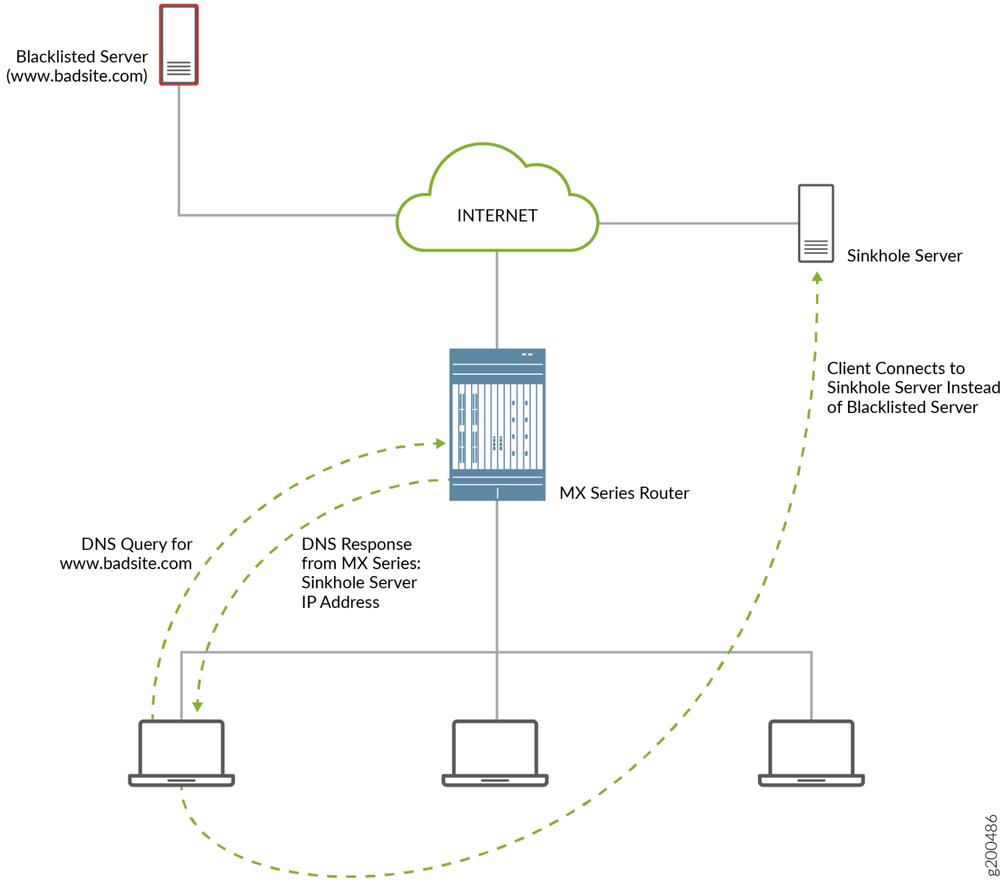
Starting in Junos OS release 21.1R1, you can also configure the following actions for a DNS request for a disallowed domain:

- Alert
- Accept
- Drop
- Drop-no-log

For other DNS request types for a disallowed domain, the request is logged and access is allowed.

The actions that the sinkhole server takes are not controlled by the DNS request filtering feature; you are responsible for configuring the sinkhole server actions. For example, the sinkhole server could send a message to the requestor that the domain is not reachable and prevent access to the disallowed domain.

Figure 50: DNS Request for Disallowed Domain



Benefits

DNS filtering redirects DNS requests for disallowed website domains to sinkhole servers, while preventing anyone operating the system from seeing the list of disallowed domains. This is because the disallowed domain names are in an encrypted format.

Disallowed Domain Filter Database File

DNS request filtering requires a disallowed domain filter database .txt file, which identifies each disallowed domain name, the action to take on a DNS request for the disallowed domain, and the IP address or fully qualified domain name (FQDN) of a DNS sinkhole server.

DNS Filter Profile

You configure a DNS filter profile to specify which disallowed domain filter database file to use. You can also specify the interfaces on which DNS request filtering is performed, limit the filtering to requests for specific DNS servers, and limit the filtering to requests from specific source IP address prefixes.

How to Configure DNS Request Filtering

IN THIS SECTION

- How to Configure a Domain Filter Database | **1092**
- How to Configure a DNS Filter Profile | **1094**
- How to Configure a Service Set for DNS Filtering | **1099**

To filter DNS requests for disallowed website domains, perform the following:

How to Configure a Domain Filter Database

Create one or more domain filter database files that include an entry for each disallowed domain. Each entry specifies what to do with a DNS request for a disallowed website domain.

To configure a domain filter database file:

1. Create the name for the file. The database file name can have a maximum length of 64 characters and must have a **.txt** extension.
2. Add a file header with a format such as **20170314_01:domain,sinkhole_ip,v6_sinkhole,sinkhole_fqdn,id,action**.

3. Add an entry in the file for each disallowed domain. You can include a maximum of 10,000 domain entries. Each entry in the database file has the following items:

   hashed-domain-name,IPv4 sinkhole address, IPv6 sinkhole address, sinkhole FQDN, ID, action

   where:

   - **hashed-domain-name** is a hashed value of the disallowed domain name (64 hexadecimal characters). The hash method and hash key that you use to produce the hashed domain value are needed when you configure DNS filtering with the Junos OS CLI.

   - **IPv4 sinkhole address** is the address of the DNS sinkhole server for IPv4 DNS requests.

   - **IPv6 sinkhole address** is the address of the DNS sinkhole server for IPv6 DNS requests.

   - **sinkhole FQDN** is the fully qualified domain name of the DNS sinkhole server.

   - **ID** is a 32-bit number that uniquely associates the entry with the hashed domain name.

   - **action** is the action to apply to a DNS request that matches the disallowed domain name. If you enter :

     - **replace**, the MX Series router sends the client a DNS response with the IP address or FQDN of the DNS sinkhole server. If you enter **report**, the DNS request is logged and then sent to the DNS server.

     - **report**, the DNS request is logged and then sent to the DNS server.

     - **alert**, the DNS request is logged and the request is sent to the DNS server.

     - **accept**, the DNS request is logged and the request is sent to the DNS server.

     - **drop**, the DNS request is dropped and the request is logged .DNS request is not sent to the DNS server.

     - **drop-no-log**, the DNS request is dropped and no syslog is generated. DNS request is not sent to the DNS server.
4. In the last line of the file, include the file hash, which you calculate by using the same key and hash method that you used to produce the hashed domain names.
5. Save the database files on the Routing Engine in the **/var/db/url-filterd** directory.
6. Validate the domain filter database file.

```
user@host> request services web-filter validate dns-filter-file-name filename hash-key key-string hash-method hash-method-name
```

7. If you make any changes to the database file, apply the changes.

```
user@host> request services web-filter update dns-filter-database filename
```

How to Configure a DNS Filter Profile

A DNS filter profile includes general settings for filtering DNS requests for disallowed website domains, and includes up to 32 templates. The template settings apply to DNS requests on specific uplink and downlink logical interfaces or routing instances, or to DNS requests from specific source IP address prefixes, and override the corresponding settings at the DNS profile level. You can configure up to eight DNS filter profiles.

To configure a DNS filter profile:

1. Configure the name for a DNS filter profile:

   ```
   [edit]
   user@host# edit services web-filter profile profile-name
   ```

   The maximum number of profiles is 8.
2. Configure the interval for logging per-client statistics for DNS filtering. The range is 0 through 60 minutes and the default is 5 minutes.

   ```
   [edit services web-filter profile profile-name]
   user@host# set global-dns-stats-log-timer minutes
   ```
3. Configure general DNS filtering settings for the profile. These values are used if a DNS request does not match a specific template.
   a. Specify the name of the domain filter database to use when filtering DNS requests.

      ```
      [edit services web-filter profile profile-name dns-filter]
      user@host# set database-file filename
      ```

   b. (Optional) To limit DNS filtering to DNS requests that are destined for specific DNS servers, specify up to three IP addresses (IPv4 or IPv6).

      ```
      [edit services web-filter profile profile-name dns-filter]
      user@host# set dns-server [ ip-address ]
      ```

c. Specify the format for the hash key.

```
[edit services web-filter profile profile-name dns-filter]
user@host# set hash-key  ascii-text
```

d. Specify the hash key that you used to create the hashed domain name in the domain filter database file.

```
[edit services web-filter profile profile-name dns-filter]
user@host# set hash-key key-string
```

e. Specify the hash method that was used to create the hashed domain name in the domain filter database file.

```
[edit services web-filter profile profile-name dns-filter]
user@host# set hash-method hash-method-name
```

The only supported hash method is `hmac-sha2-256`.

f. Configure the interval for logging statistics for DNS requests and for sinkhole actions performed for each customer IP address. The range is 1 through 60 minutes and the default is 5 minutes.

```
[edit services web-filter profile profile-name dns-filter]
user@host# set statistics-log-timer minutes
```

g. Configure the time to live while sending the DNS response after taking the DNS sinkhole action. The range is 0 through 86,400 seconds and the default is 1800.

```
[edit services web-filter profile profile-name dns-filter]
user@host# set dns-resp-ttl seconds
```

h. Configure the level of subdomains that are searched for a match. The range is 0 through 10. A value of 0 indicates that subdomains are not searched.

```
[edit services web-filter profile profile-name dns-filter]
user@host# set wildcarding-level level
```

For example, if you set the `wildcarding-level` to 4 and the database file includes an entry for **example.com**, the following comparisons are made for a DNS request that arrives with the domain **198.51.100.0.example.com**:

- **198.51.100.0.example.com**: no match
- **51.100.0.example.com**: no match for one level down
- **100.0.example.com**: no match for two levels down
- **0.example.com**: no match for three levels down
- **example.com**: match for four levels down

4. Configure a template. You can configure a maximum of 8 templates in a profile. Each template identifies filter settings for DNS requests on specific uplink and downlink logical interfaces or routing instances, or for DNS requests from specific source IP address prefixes.

   a. Configure the name for the template.

   ```
   [edit services web-filter profile profile-name]
   user@host# set dns-filter-template template-name
   ```

   b. (Optional) Specify the client-facing logical interfaces (uplink) to which the DNS filtering is applied.

   ```
   [edit services web-filter profile profile-name dns-filter-template template-name]
   user@host# set client-interfaces client-interface-name
   ```

   c. (Optional) Specify the server-facing logical interfaces (downlink) to which the DNS filtering is applied.

   ```
   [edit services web-filter profile profile-name dns-filter-template template-name]
   user@host# set server-interfaces server-interface-name
   ```

   d. (Optional) Specify the routing instance for the client-facing logical interface to which the DNS filtering is applied.

   ```
   [edit services web-filter profile profile-name dns-filter-template template-name]
   user@host# set client-routing-instance client-routing-instance-name
   ```

e. (Optional) Specify the routing instance for the server-facing logical interface to which DNS filtering is applied.

```
[edit services web-filter profile profile-name dns-filter-template template-name]
user@host# set server-routing-instance server-routing-instance-name
```

NOTE: If you configure the client and server interfaces or the client and server routing instances, implicit filters are installed on the interfaces or routing instances to direct DNS traffic to the services PIC for DNS filtering. If you configure neither the client and server interfaces nor the routing instances, you must provide a way to direct DNS traffic to the services PIC (for example, via routes).

f. Specify the name of the domain filter database to use when filtering DNS requests.

```
[edit services web-filter profile profile-name dns-filter-template template-name dns-
filter]
user@host# set database-file filename
```

g. (Optional) To limit DNS filtering to DNS requests that are destined for specific DNS servers, specify up to three IP addresses (IPv4 or IPv6).

```
[edit services web-filter profile profile-name dns-filter-template template-name dns-
filter]
user@host# set dns-server ip-address
```

h. Specify the hash method that was used to create the hashed domain name in the domain filter database file.

```
[edit services web-filter profile profile-name dns-filter-template template-name dns-
filter]
user@host# set hash-method hash-method-name
```

The only supported hash method is `hmac-sha2-256`.

i. Specify the hash key that was used to create the hashed domain name in the domain filter database file.

```
[edit services web-filter profile profile-name dns-filter-template template-name dns-
filter]
user@host# set hash-key key-string
```

j. Configure the interval for logging statistics for DNS requests and for sinkhole actions performed for each customer IP address. The range is 1 through 60 minutes and the default is 5 minutes.

```
[edit services web-filter profile profile-name dns-filter-template template-name dns-
filter]
user@host# set statistics-log-timer minutes
```

k. Configure the time to live while sending the DNS response after taking the DNS sinkhole action. The range is 0 through 86,400 seconds and the default is 1800.

```
[edit services web-filter profile profile-name dns-filter-template template-name dns-
filter]
user@host# set dns-resp-ttl seconds
```

l. Configure the level of subdomains that are searched for a match. The range is 0 through 10. A value of 0 indicates that subdomains are not searched.

```
[edit services web-filter profile profile-name dns-filter-template template-name dns-
filter]
user@host# set wildcarding-level level
```

For example, if you set the `wildcarding-level` to 4 and the database file includes an entry for **example.com**, the following comparisons are made for a DNS request that arrives with the domain **198.51.100.0.example.com**:

- **198.51.100.0.example.com**: no match
- **51.100.0.example.com**: no match for one level down
- **100.0.example.com**: no match for two levels down
- **0.example.com**: no match for three levels down
- **example.com**: match for four levels down

m. (Optional) Specify the response error code for SRV and TXT query types.

(Optional) Specify the response error code for SRV and TXT query types.

```
[edit services web-filter profile profile-name dns-filter-template template-name dns-
filter]
user@host# set txt-resp-err-code (Noerror | Refused)
user@host# set srv-resp-err-code (Noerror | Refused)
```

n. Configure a term for the template. You can configure a maximum of 64 terms in a template.

```
[edit services web-filter profile profile-name dns-filter-template template-name]
user@host# set term term-name
```

o. (Optional) Specify the source IP address prefixes of DNS requests you want to filter. You can configure a maximum of 64 prefixes in a term.

```
[edit services web-filter profile profile-name dns-filter-template template-name term term-
name]
user@host# set from src-ip-prefix source-prefix
```

p. Specify that the sinkhole action identified in the domain filter database is performed on disallowed DNS requests.

```
[edit services web-filter profile profile-name dns-filter-template template-name term term-
name]
user@host# set then dns-sinkhole
```

How to Configure a Service Set for DNS Filtering

- Associate the DNS filter profile with a next-hop service set and enable logging for DNS filtering. The service interface can be an ms- or vms- interface Next Gen Services with MX-SPC3 services card), or it can be an aggregated multiservices (AMS) interface.

```
[edit services service-set service-set-name]
user@host# set web-filter-profile profile-name
user@host# set syslog host hostname class urlf-logs
```

```
user@host# set next-hop-service inside-service-interface interface-name.unit-number
user@host# set next-hop-service outside-service-interface interface-name.unit-number
```

Multitenant Support for DNS Filtering

IN THIS SECTION

- Overview | 1100

Overview

Starting in Junos OS Release 21.1R1, you can configure custom domain feeds per customer or IP subgroup. You can :

- Configure domain names and actions for multiple tenants such that domain feeds can be managed on a per tenant basis.
- Configure hierarchical domain feed management per profile, per dns-filter-template or per dns-filter-term.
- Exempt domain feeds at the IP, subnet, or CIDR level.

To implement the mutiltenant support for DNS filtering, creating the domain filter database file under template or profile level is disabled. You need not specify a file at the template or profile level. Starting in Junos OS 21.1R1, by default, a global file with a fixed name, **nsf_multi_tenant_dn_custom_file.txt** (plain text format) or **dnsf_multi_tenant_dn_custom_file_hashed.txt** (encrypted file) is available.

Each entry in the database file has the following items:

hashed-domain-name, IPv4 sinkhole address, IPv6 sinkhole address, sinkhole FQDN, ID, action, feed-name.

The file hash is calculated and appended to the list of domain name entries in the file. The file hash is calculated using a global key and method ,which is validated with the file hash computed using the hash key configured at the `[edit services web-filter]` hierarchy. The file validation is successful only if the calculated file-hash matches the file hash present in the file.

Each entry in **nsf_multi_tenant_dn_custom_file.txt** file consists of an additional field called **feed-name**. This **feed-name** s used as an indicator to group set of domain-names and map them to a tenant (profile, template, term, or IP address).

When the DNS packets are received from a particular SRC IP address, the corresponding feed-name is fetched and lookup happens against the domain-names mapped with the feed-name associated with the term. If the feed-name is not provisioned for that IP address, then it falls back to the feed-name configured at the template-level and lookup happens against the domain-names mapped with the feed-name associated with the template. If the feed-name is not configured at template, then the lookup is against the domain-names mapped against the feed-name associated with the profile.

Configuring Multi-tenant Support for DNS Filtering

1. Configure the web filter.

```
[edit]
user@host# edit services web-filter
```

2. Enable multi-tenant support

```
[edit services web-filter]
user@host# set multi-tenant-support
```

3. Configure the global file hash key and hash method.

```
[edit services web-filter]
user@host# set multi-tenant-hash
user@host# set multi-tenant-hash file-hash-key (ascii-text | hexadecimal)
user@host# set multi-tenant-hash hash-method (ascii-text | hexadecimal)
```

> **NOTE**: When `multi-tenant-hash`is configured, it indicates that the global dns feed file consists of only encrypted feeds. When `multi-tenant-hash` s not configured it indicates that the global dns feed file has feeds in plain text format.

4. Configure the name for a DNS filter profile and map the domain feed at the profile level. The feed name indicator configured at the profile level is applied to all the templates and terms under the profile that do not have the feed name indicator configured.

```
[edit]
user@host# [edit services web-filter profile profile-name]
user@host# [edit services web-filter profile profile-name feed-name feed-name]
```

5. Configure general DNS filtering settings for the profile. These values are used if a DNS request does not match a specific template.

   a. (Optional) To limit DNS filtering to DNS requests that are destined for specific DNS servers, specify up to three IP addresses (IPv4 or IPv6).

   ```
   [edit services web-filter profile profile-name dns-filter]
   user@host# set dns-server [ip-address]
   ```

   b. Configure the interval for logging statistics for DNS requests and for sinkhole actions performed for each customer IP address. The range is 1 through 60 minutes and the default is 5 minutes.

   ```
   [edit services web-filter profile profile-name dns-filter]
   user@host# set statistics-log-timer minutes
   ```

   c. Configure the time to live (TTL) to send the DNS response after taking the DNS sinkhole action. The range is 0 through 86,400 seconds and the default is 1800.

   ```
   [edit services web-filter profile profile-name dns-filter]
   user@host# set  dns-resp-ttlseconds
   ```

   d. Configure the level of subdomains that are searched for a match. The range is 0 through 10. A value of 0 indicates that subdomains are not searched.

   ```
   [edit services web-filter profile profile-name dns-filter]
   user@host# set wildcarding-levellevel
   ```

   e. (Optional) Specify the response error code for the TXT query type.

   ```
   [edit services web-filter profile profile-name dns-filter]
   user@host# set txt-resp-err-code (Noerror | Refused) level
   ```

6. Configure a template. You can configure a maximum of 8 templates in a profile. Each template identifies filter settings for DNS requests on specific uplink and downlink logical interfaces or routing instances, or for DNS requests from specific source IP address prefixes.

a. Configure the name for the template.

```
[edit services web-filter profile profile-name]
user@host# set dns-filter-template template-name
```

b. Configure the feed name. With multitenant format, you can no longer add a file name under profile or template. The feed name specified under profile has lesser precedence compared to the one configured under the template.

```
[edit services web-filter profile profile-namedns-filter-template template-name ]
user@host# set feed-name feed-name
```

c. (Optional) Specify the client-facing logical interfaces (uplink) to which the DNS filtering is applied.

```
[edit services web-filter profile profile-name dns-filter-template template-name]
user@host# set client-interfaces client-interface-name
```

d. (Optional) Specify the server-facing logical interfaces (downlink) to which the DNS filtering is applied.

```
[edit services web-filter profile profile-name dns-filter-template template-name]
user@host# set server-interfaces server-interface-name
```

e. (Optional) Specify the routing instance for the client-facing logical interface to which the DNS filtering is applied.

```
[edit services web-filter profile profile-name dns-filter-template template-name]
user@host# set client-routing-instance client-routing-instance-name
```

f. (Optional) Specify the routing instance for the server-facing logical interface to which DNS filtering is applied.

```
[edit services web-filter profile profile-name dns-filter-template template-name]
user@host# set server-routing-instance server-routing-instance-name
```

NOTE: If you configure the client and server interfaces or the client and server routing instances, implicit filters are installed on the interfaces or routing instances to direct DNS traffic to the services PIC for DNS filtering. If you configure neither the client and server interfaces nor the routing instances, you must provide a way to direct DNS traffic to the services PIC (for example, through routes).

g. Configure the interval for logging statistics for DNS requests and for sinkhole actions performed for each customer IP address. The range is 1 through 60 minutes and the default is 5 minutes.

```
[edit services web-filter profile profile-name dns-filter-template template-name dns-
filter]
user@host# set statistics-log-timer minutes
```

h. Configure the time to live while sending the DNS response after taking the DNS sinkhole action. The range is 0 through 86,400 seconds and the default is 1800.

```
[edit services web-filter profile profile-name dns-filter-template template-name dns-
filter]
user@host# set dns-resp-ttl seconds
```

i. Configure the level of subdomains that are searched for a match. The range is 0 through 10. A value of 0 indicates that subdomains are not searched.

```
[edit services web-filter profile profile-name dns-filter-template template-name dns-
filter]
user@host# set wildcarding-level level
```

j. Configure a term for the template. You can configure a maximum of 64 terms in a template.

```
[edit services web-filter profile profile-name dns-filter-template template-name]
user@host# set term term-name
```

k. Configure the feed name. The feed name configured at the term takes higher precedence over the one configured under the template. However, if the sinkhole domain is matching the only domain mentioned in the feed name under template, the action specified for that entry is implemented.

```
[edit services web-filter profile profile-name dns-filter-template template-name term term-
name]
user@host# set feed-name feed-name
```

l. (Optional) Specify the source IP address prefixes of DNS requests you want to filter. You can configure a maximum of 64 prefixes in a term.

```
[edit services web-filter profile profile-name dns-filter-template template-name term term-
name]
user@host# set from src-ip-prefix source-prefix
```

m. Configure that the sinkhole action identified in the domain filter database is performed on disallowed DNS requests.

```
[edit services web-filter profile profile-name dns-filter-template template-name term term-
name]
user@host# set then dns-sinkhole
```

7. Associate the DNS filter profile with a next-hop service set and enable logging for DNS filtering. The service interface can be a multiservices (ms) or virtual multi service (vms) interface (Next Gen Services with MX-SPC3 services card), or it can be an aggregated multiservices (AMS) interface.

```
[edit services service-set service-set-name]
user@host# set syslog mode event
user@host# set syslog syslog event-rate vent-rate
user@host# set syslog local-category urlf
user@host# set web-filter-profile profile-name
user@host# set set next-hop-service inside-service-interface interface-name.unit-number
user@host# set set next-hop-service outside-service-interface interface-name.unit-number
```

8. If you are running Next Gen Services on the MX-SPC3 services card, configure the vms interface to get the FPC and PIC information in the syslog.

```
[edit interfaces interface-name]
user@host# set vms 0/0/0
user@host# set services-options
```

```
[edit interfaces interface-name
user@host# fpc-pic-information
```

Example: Configuring Multitenant Support for DNS Filtering

IN THIS SECTION

- Configuration | **1106**

Configuration

IN THIS SECTION

- CLI Quick Configuration | **1106**

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the [edit] hierarchy level.

```
set services service-set Test_Zone3 syslog mode stream
set services service-set Test_Zone3 syslog source-address 10.1.1.1
set services service-set Test_Zone3 syslog stream t1 category urlf
set services service-set Test_Zone3 syslog stream t1 host 10.10.1.1
set services service-set Test_Zone3 syslog stream t1 routing-instance client_vr4
set services service-set Test_Zone3 web-filter-profile Test-Profile-3-Zone3
set services service-set Test Zone3 next-hop-service inside-service-interface ams3.24
set services service-set Test Zone3 next-hop-service outside-service-interface ams3.25
set services web-filter multi-tenant-support
set services web-filter  multi-tenant-hash file-hash-key ascii-text "$9$VjsgJikP36AGD6Ap0hcbs2"
set services web-filter  multi-tenant-hash hash-method hmac-sha2-256
set services web-filter profile Test-Profile-3-Zone3 feed-name abc
set services web-filter profile Test-Profile-3-Zone3 global-dns-filter-stats-log-timer 20
set services web-filter profile Test-Profile-3-Zone3 dns-filter statistics-log-timer 5
set services web-filter profile Test-Profile-3-Zone3 dns-filter dns-resp-ttl 100
set services web-filter profile Test-Profile-3-Zone3 dns-filter wildcarding-level 10
 set services web-filter profile Test-Profile-3-Zone3 dns-filter-template Test-Profile-3-Zone3-
Area1 inactive: client-interfaces xe-7/0/2.32
set services web-filter profile Test-Profile-3-Zone3 dns-filter-template Test-Profile-3-Zone3-
Area1 inactive: server-interfaces xe-7/2/0.36
set services web-filter profile Test-Profile-3-Zone3 dns-filter-template Test-Profile-3-Zone3-
Area1 inactive: client-routing-instance client_vr4
set services web-filter profile Test-Profile-3-Zone3 dns-filter-template Test-Profile-3-Zone3-
Area1 inactive: server-routing-instance server_vr4
set services web-filter profile Test-Profile-3-Zone3 dns-filter-template Test-Profile-3-Zone3-
Area1 term Test-Profile-3-Zone3-Area1-Customer1 feed-name customer2
set services web-filter profile Test-Profile-3-Zone3 dns-filter-template Test-Profile-3-Zone3-
Area1 term Test-Profile-3-Zone3-Area1-Customer1 from src-ip-prefix 10.12.1.1
set services web-filter profile Test-Profile-3-Zone3 dns-filter-template Test-Profile-3-Zone3-
Area1 term Test-Profile-3-Zone3-Area1-Customer1 then dns-sinkhole
set services web-filter profile Test-Profile-3-Zone3 dns-filter-template Test-Profile-3-Zone3-
Area1 term Test-Profile-3-Zone3-Area1-Customer2 feed-name customer2
set services web-filter profile Test-Profile-3-Zone3 dns-filter-template Test-Profile-3-Zone3-
Area1 term Test-Profile-3-Zone3-Area1-Customer2 from src-ip-prefix 2001:db8::0/96
 set services web-filter profile Test-Profile-3-Zone3 dns-filter-template Test-Profile-3-Zone3-
Area1 term Test-Profile-3-Zone3-Area1-Customer2 then dns-sinkhole
set services web-filter profile Test-Profile-3-Zone3 dns-filter-template Test-Profile-3-Zone3-
Area1 term Test-Profile-3-Zone3-Area1-Customer3 from src-ip-prefix 2001:db8:bbbb::/96
set services web-filter profile Test-Profile-3-Zone3 dns-filter-template Test-Profile-3-Zone3-
Area1 term Test-Profile-3-Zone3-Area1-Customer3 then dns-sinkhole
set services web-filter profile Test-Profile-3-Zone3 dns-filter-template Test-Profile-3-Zone3-
Area2 inactive: client-interfaces xe-7/0/2.32
set services web-filter profile Test-Profile-3-Zone3 dns-filter-template Test-Profile-3-Zone3-
Area2 inactive: server-interfaces xe-7/2/0.36
set services web-filter profile Test-Profile-3-Zone3 dns-filter-template Test-Profile-3-Zone3-
Area2 inactive: client-routing-instance client_vr4
set services web-filter profile Test-Profile-3-Zone3 dns-filter-template Test-Profile-3-Zone3-
Area2 inactive: server-routing-instance server_vr4
set services web-filter profile Test-Profile-3-Zone3 dns-filter-template Test-Profile-3-Zone3-
```

```
Area2 term Test-Profile-3-Zone3-Area2-Customer1 from src-ip-prefix 22.21.128.0/17
set services web-filter profile Test-Profile-3-Zone3 dns-filter-template Test-Profile-3-Zone3-
Area2 term Test-Profile-3-Zone3-Area2-Customer1 then dns-sinkhole
set services web-filter profile Test-Profile-3-Zone3 dns-filter-template Test-Profile-3-Zone4-
Area2 feed-name customer2
set services web-filter profile Test-Profile-3-Zone3 dns-filter-template Test-Profile-3-Zone4-
Area2 inactive: client-routing-instance client_vr4
set services web-filter profile Test-Profile-3-Zone3 dns-filter-template Test-Profile-3-Zone4-
Area2 inactive: server-routing-instance server_vr4
set services web-filter profile Test-Profile-3-Zone3 dns-filter-template Test-Profile-3-Zone4-
Area2 term Test-Profile-3-Zone4-Area2-Customer1 from src-ip-prefix 2001:0db8:0001:/48
set services web-filter profile Test-Profile-3-Zone3 dns-filter-template Test-Profile-3-Zone4-
Area2 term Test-Profile-3-Zone4-Area2-Customer1 then dns-sinkhole
set services web-filter profile Test-Profile-3-Zone3 dns-filter-template Test-Profile-3-Zone4-
Area2 term wildcard then dns-sinkhole
set interfaces xe-7/0/0 unit 0 family inet address 10.11.1.1/24
set interfaces xe-7/0/1 unit 0 family inet address 10.12.1.1/24
set interfaces xe-7/0/2 flexible-vlan-tagging
set interfaces xe-7/0/2 mtu 9192
set interfaces xe-7/0/2 encapsulation flexible-ethernet-services
set interfaces xe-7/0/2 unit 1 vlan-id 10
set interfaces xe-7/0/2 unit 1 family inet address 198.31.100.1/24
set interfaces xe-7/0/2 unit 31 vlan-id 31
set interfaces xe-7/0/2 unit 31 family inet address 198.51.70.1/24;
set interfaces xe-7/0/2 unit 31 family inet6 address 2001:db8:10::0/96
set interfaces xe-7/0/2 unit 32 vlan-id 32
set interfaces xe-7/0/2 unit 32 family inet address 198.51.71.1/24;
set interfaces xe-7/0/2 unit 32 family inet6 address 2001:db8:11::0/96
set interfaces xe-7/0/2 unit 33 vlan-id 33
set interfaces xe-7/0/2 unit 33 family inet address 198.51.72.1/24
set interfaces xe-7/0/2 unit 33 family inet6 address 2001:db8:12::0/96
set interfaces xe-7/0/2 unit 34 vlan-id 34
set interfaces xe-7/0/2 unit 34 family inet address 198.51.73.1/24
set interfaces xe-7/0/2 unit 34 family inet6 address 2001:db8:13::0/96
set interfaces xe-7/0/2 unit 35 vlan-id 35
set interfaces xe-7/0/2 unit 35 vlan-id 35 family inet address 198.51.74.1/24
set interfaces xe-7/0/2 unit 3135 vlan-id 35 family inet6 address 2001:db8:14::0/96
set interfaces xe-7/0/2 unit 36 vlan-id 36
set interfaces xe-7/0/2 unit 36 family inet address 198.51.75.1/24
set interfaces xe-7/0/2 unit 36 family inet6 address 2001:db8:15::0/96
set interfaces xe-7/0/2 unit 37 vlan-id 37
set interfaces xe-7/0/2 unit 37 family inet address 198.51.76.1/24
set interfaces xe-7/0/2 unit 37 family inet6 address 2001:db8:16::0/96
set interfaces xe-7/0/2 unit 38 vlan-id 38
set interfaces xe-7/0/2 unit 38 family inet address 198.51.77.1/24
set interfaces xe-7/0/2 unit 38 family inet6 address 2001:db8:17::0/96
set interfaces xe-7/0/2 unit 39 vlan-id 39
set interfaces xe-7/0/2 unit 39 family inet address 198.51.78.1/24
set interfaces xe-7/0/2 unit 39 family inet6 address 2001:db8:18::0/96
set interfaces xe-7/0/2 unit 40 vlan-id 40
set interfaces xe-7/0/2 unit 40 family inet address 198.51.79.1/24
set interfaces xe-7/0/2 unit 40 family inet6 address 2001:db8:19::0/96
set interfaces xe-7/0/2 unit 41 vlan-id 41
set interfaces xe-7/0/2 unit 41 family inet address 198.51.80.1/24
set interfaces xe-7/0/2 unit 41 family inet6 address 2001:db8:20::0/96
set interfaces xe-7/2/0 flexible-vlan-tagging
set interfaces xe-7/2/0 mtu 1514
set interfaces xe-7/2/0 encapsulation flexible-ethernet-services
set interfaces xe-7/2/0 inactive unit 1 vlan-id 1
set interfaces xe-7/2/0 inactive unit 1 family inet address 198.168.50.0/24
set interfaces xe-7/2/0 inactive unit 1 family inet6 address 2001:0db0:1600:0::1/112
set interfaces xe-7/2/0 unit 2 vlan-id 2
set interfaces xe-7/2/0 unit 2 vlan-id 2 family inet address 198.100.70.0/24
set interfaces xe-7/2/0 unit 31 vlan-id 31
set interfaces xe-7/2/0 unit 31 family inet address 10.1.0.1/16
set interfaces xe-7/2/0 unit 31 family inet6 address 2001:0db0:1601:0::1/112
set interfaces xe-7/2/0 unit 32 vlan-id 32
set interfaces xe-7/2/0 unit 32 family inet address 10.2.0.1/16
set interfaces xe-7/2/0 unit 32 family inet6 address 2001:0db0:1602:0::1/112
set interfaces xe-7/2/0 unit 33 vlan-id 33
set interfaces xe-7/2/0 unit 33 family inet address 10.3.0.1/16
set interfaces xe-7/2/0unit 33 vlan-id 33 family inet6 address 2001:0db0:1603:0::1/112
set interfaces xe-7/2/0 unit 34 vlan-id 34
set interfaces xe-7/2/0 unit 34 family inet address 10.0.0.1/16
set interfaces xe-7/2/0 unit 34 family inet6 address 2001:0db0:1600:0::1/112
set interfaces xe-7/2/0 unit 35 vlan-id 35
set interfaces xe-7/2/0 unit 35 family inet address 10.4.0.1/16
set interfaces xe-7/2/0 unit 35 family inet6 address 2001:0db0:1604:0::1/112
set interfaces xe-7/2/0 unit 36 vlan-id 36
set interfaces xe-7/2/0 unit 36 family inet address 10.5.0.1/16
set interfaces xe-7/2/0 unit 36 family inet6 address 2001:0db0:1605:0::1/112
set interfaces xe-7/2/0 unit 37 vlan-id 37
set interfaces xe-7/2/0 unit 37 family inet address 10.6.0.1/16
set interfaces xe-7/2/0unit 37 family inet6 address 2001:0db0:1606:0::1/112
set interfaces xe-7/2/0 unit 38 vlan-id 38
set interfaces xe-7/2/0 unit 38 family inet address 10.7.0.1/16
set interfaces xe-7/2/0 unit 38 vlan-id 38 family inet6 address 2001:0db0:160:0::1/112
set interfaces ams3  load-balancing-options member-interface mams-3/0/0
set interfaces ams3  load-balancing-options member-interface mams-3/1/0
set interfaces ams3  load-balancing-options member-failure-options redistribute-all-traffic
enable-rejoin
set interfaces ams3  load-balancing-options high-availability-options many-to-one preferred-
backup mams-3/1/0
set interfaces ams3  unit 22 family inet
set interfaces ams3  unit 22 family inet6
set interfaces ams3  unit 22 service-domain inside
set interfaces ams3  unit 22 load-balancing-options hash-keys ingress-key (source-ip destination-
ip )
set interfaces ams3  unit 24 family inet
set interfaces ams3  unit 24 family inet6
set interfaces ams3  unit 24 service-domain inside
set interfaces ams3  unit 24 family inet6 load-balancing-options hash-keys ingress-key (source-
ip destination-ip)
set interfaces ams3  unit 25 family inet
set interfaces ams3  unit 25 family inet6
set interfaces ams3  unit 25 service-domain inside
set interfaces ams3  unit 25 load-balancing-options hash-keys ingress-key (source-ip destination-
ip )
set routing-instances client_vr4 instance-type virtual-router
set routing-instances client_vr4 routing-options rib client_vr4.inet6.0 static route
2001:0db0:bbbb:0::0/49 next-hop 2001:0db0:7070:71::2
set routing-instances client_vr4 routing-options rib client_vr4.inet6.0 static route
2001:0db0:aaaa:8000::0/49 next-hop 2001:0db0:7070:71::3
set routing-instances client_vr4 routing-options rib client_vr4.inet6.0 static route 60::0/64
next-hop ams3.24
set routing-instances client_vr4 routing-options static route 10.12.1.1 next-hop 192.168.1.2
set routing-instances client_vr4 routing-options static route 22.21.128.0/17 next-hop 192.168.1.3
set routing-instances client_vr4 routing-options static route 0.0.0.0/0 next-hop ams3.24
set routing-instances client_vr4 routing-options static route 10.11.10.10/16 next-hop 192.168.1.4
set routing-instances client_vr4 routing-options static route 10.10.23.10/16 next-hop 192.168.1.5
set routing-instances client_vr4 routing-options static route 10.1.0.0/16 next-hop 192.168.1.6
set routing-instances client_vr4 routing-options static route 10.20.20.0/16 next-hop 192.168.1.7
set routing-instances client_vr4 routing-options static route 10.2.0.0/16 next-hop 192.168.1.8
set routing-instances client_vr4 routing-options static route 10.30.20.0/16 next-hop 192.168.1.9
set routing-instances client_vr4 routing-options static route 10.3.0.0/16 next-hop 192.168.10.
set routing-instances client_vr4 routing-options static route 10.40.20.0/16 next-hop 192.168.1.11
set routing-instances client_vr4 routing-options static route 10.4.0.0/16 next-hop 192.168.1.12
set routing-instances client_vr4 routing-options static route 10.50.20.0/16 next-hop 192.168.1.13
set routing-instances client_vr4 interface xe-7/0/0.0
set routing-instances client_vr4 interface xe-7/0/2.32
set routing-instances client_vr4 interface ams3.24
set routing-instances server_vr4 instance-type virtual-router
set routing-instances server_vr4 routing-options rib server_vr4.inet6.0 static route
2001:0db0:2221:0::0/48 next-hop ams3.25
set routing-instances server_vr4 routing-options rib server_vr4.inet6.0 static route
2001:db8:ffff::1/128 next-hop 2001:0db0:1605:0::2
set routing-instances server_vr4 routing-options rib server_vr4.inet6.0 static route
2001:db8:bbbb::1/128 next-hop 2001:0db0:1605:0::3
set routing-instances server_vr4 routing-options static route 10.10.20.1 next-hop ams3.25
set routing-instances server_vr4 routing-options static route 60.0.6.0/24 next-hop 192.0.2.2
set routing-instances server_vr4 routing-options static route 60.0.18.0/24 next-hop 192.0.2.3
set routing-instances server_vr4 routing-options static route 10.9.9.0/24 next-hop ams3.25
set routing-instances server_vr4 routing-options static route 60.0.19.0/24 next-hop 192.0.2.4
set routing-instances server_vr4 routing-options static route 60.0.20.0/24 next-hop 192.0.2.5
set routing-instances server_vr4 routing-options static route 60.0.21.0/24 next-hop 192.0.2.6
set routing-instances server_vr4 routing-options static route 60.0.22.0/24 next-hop 192.0.2.7
set routing-instances server_vr4 routing-options static route 60.0.23.0/24 next-hop 192.0.2.8
set routing-instances server_vr4 routing-options static route 60.0.24.0/24 next-hop 192.0.2.9
set routing-instances server_vr4 routing-options static route 60.0.25.0/24 next-hop 192.0.2.10
set routing-instances server_vr4 routing-options static route 60.0.26.0/24 next-hop 192.0.2.11
set routing-instances server_vr4 routing-options static route 60.0.27.0/24 next-hop 192.0.2.12
set routing-instances server_vr4 routing-options static route 60.0.28.0/24 next-hop 192.0.2.13
set routing-instances server_vr4 routing-options static route 10.1.0.0/16 next-hop ams3.25
set routing-instances server_vr4 interface xe-7/0/1.0
set routing-instances server_vr4 interface xe-7/2/0.36
set routing-instances server_vr4 interface ams3.25
set routing-options static route 0.0.0.0/0 next-hop 10.48.179.254
```

Integration of Juniper ATP Cloud and Web Filtering on MX Series Routers

IN THIS SECTION

- Overview | 1112
- Configuring the Web Filter Profile for Sampling | 1116
- GeoIP Filtering | 1121
- Global Allowlist and Global Blocklist | 1123

Overview

IN THIS SECTION

- Benefits | 1112
- Understanding Policy Enforcer and Juniper ATP Cloud | 1112
- Security Intelligence (SecIntel) - Overview | 1114
- Web Filtering (URL-Filterd) - Overview | 1115

Juniper Advanced Threat Prevention (Juniper ATP Cloud) is integrated with MX series routers to protect all hosts in your network against evolving security threats by employing cloud-based threat detection software with a next-generation firewall system.

This topic provides an overview of Juniper ATP Cloud, Policy Enforcer, Security Intelligence, Web filtering, and their benefits when integrated on MX Series routers (MX240, MX480 and MX960).

Benefits

- Simplifies deployment and enhances the anti-threat capabilities when integrated with the MX routers.
- Delivers protection against “zero-day” threats using a combination of tools to provide robust coverage against sophisticated, evasive threats.
- Checks inbound and outbound traffic with policy enhancements that allow users to stop malware, quarantine infected systems, prevent data exfiltration, and disrupt lateral movement.
- Supports High Availability to provide uninterrupted service.
- Provides scalability to handle increasing loads that require more computing resources, increased network bandwidth to receive more customer submissions, and a large storage for malware.
- Provides deep inspection, actionable reporting, and inline malware blocking.

Understanding Policy Enforcer and Juniper ATP Cloud

Juniper Networks Security Director comprises a feature called the Policy Enforcer (PE) that enables it to learn from threat conditions, automate the policy creation, and to dynamically deploy enforcement to Juniper devices in the network.

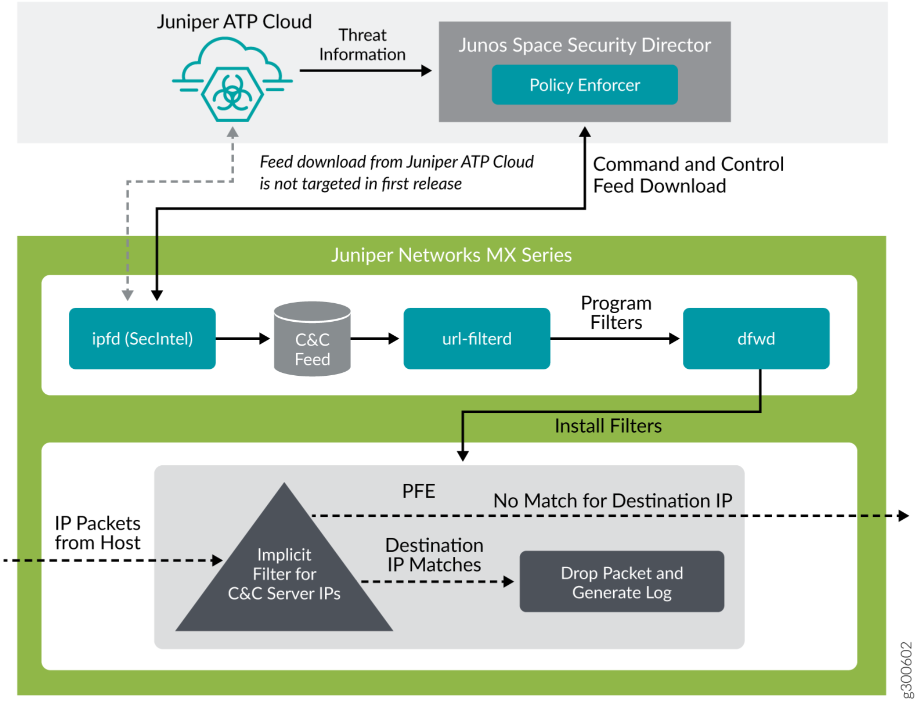
Figure 4 illustrates the traffic flow between the PE, the Juniper ATP Cloud, and the MX router which functions as a firewall.

- Policy Enforcer (PE) learns from threat conditions, automates the policy creation, and deploys enforcement to Juniper devices in the network.

- Juniper Advanced Threat Prevention (Juniper ATP Cloud) protects all hosts in your network by employing cloud-based threat detection software with a next-generation firewall system.

- MX router fetches the threat intelligence feeds from Policy Enforcer (PE) and implements those policies to quarantine compromised hosts. It comprises of the following important components:
  - Security Intelligence process
  - Web Filtering process
  - Firewall process

Figure 51: System Architecture


Juniper ATP Cloud
Threat Information
Junos Space Security Director
Policy Enforcer
Feed download from Juniper ATP Cloud is not targeted in first release
Command and Control Feed Download
Juniper Networks MX Series
ipfd (SecIntel)
C&C Feed
url-filterd
Program Filters
dfwd
Install Filters
PFE
No Match for Destination IP
IP Packets from Host
Implicit Filter for C&C Server IPs
Destination IP Matches
Drop Packet and Generate Log
g300602

To understand the functionality of the system architecture consider the following example—if a user downloads a file from the Internet and that file passes through an MX firewall, the file can be sent to the

Juniper ATP Cloud cloud for malware inspection (depending on your configuration settings.) If the file is determined to be malware, PE identifies the IP address and MAC address of the host that downloaded the file. Based on a user-defined policy, that host can be put into a quarantine VLAN or blocked from accessing the Internet.

MX Series routers (MX240, MX480, and MX960) can be integrated with the Juniper ATP Cloud to prevent compromised hosts (botnets) from communicating with command and control servers:

- Starting in Junos OS Release 18.4R1 with the Adaptive Services as an Inline security capability
- Starting in Junos OS Release 19.3R2 with the Next Gen Services as an Inline security capability

Security Intelligence (SecIntel) - Overview

The Security Intelligence process (IPFD), is responsible for downloading the security intelligence feeds and parsing from the feed connector or ATP Cloud cloud feed server. The IPFD process on the MX platforms fetches the command and control IPv4/IPv6 feeds from Policy Enforcer. C&C feeds are essentially a list of servers that are known command and control servers for botnets. The list also includes servers that are known sources for malware downloads. The information thus fetched is saved in a file (**urlf_si_cc_db.txt**) created under the **/var/db/url-filterd** directory.

The file format of the disallowed IPs sent by IPFD to the web filtering process is as follows:

`IPv4 address | IPv6 address, threat-level.`

The `threat-level` is an integer ranging from 1 to 10 to indicate the threat level of files scanned for malware and for infected hosts. Here, 1 represents the lowest threat level and 10 represents the highest threat level.

For example: 178.10.19.20, 4

Here, 178.10.19.20 indicates the disallowed IP and 4 indicates the `threat-level`.

The C&C feed database is synced onto the backup Routing Engine. IPFD then shares the information to the web filtering process (url-filterd). The web filtering process reads the file contents and configures the filters accordingly.

Configuring Security Intelligence to Download the CC Feed from Policy Enforcer

To download the command and control IPv4/IPv6 feeds from Juniper ATP Cloud/Policy Enforcer, include the `security-intelligence` statement at the `[edit services]` hierarchy as shown in the following example:

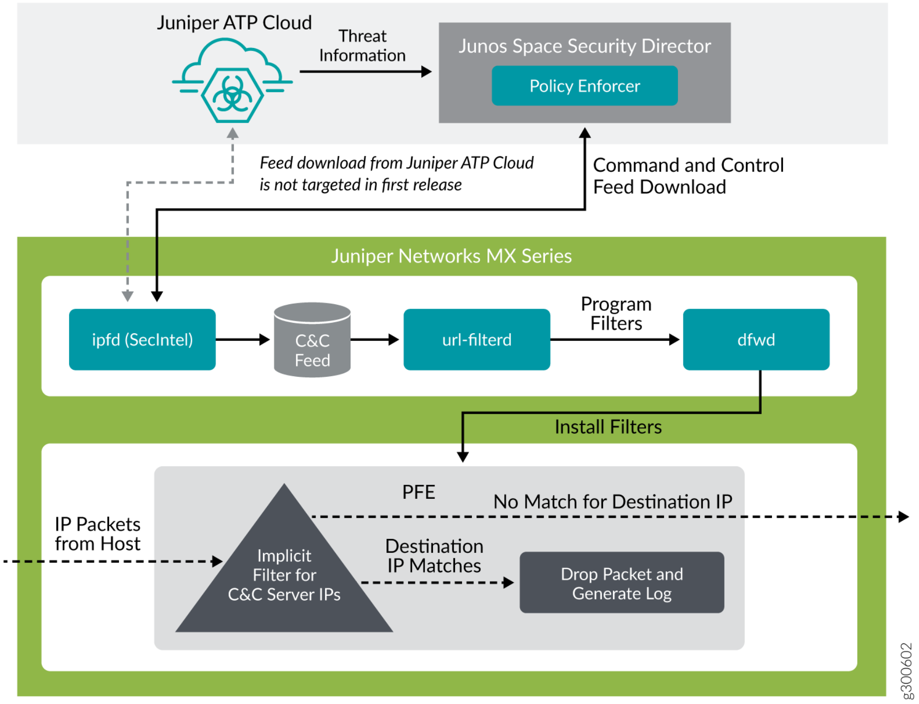
```
security-intelligence {
    authentication {
            auth-token 7QGSBL5ZRKR5UHUZ2X2R6QLHB656D5EN;
        }
        url https://10.92.83.245:443/api/v1/manifest.xml;
        traceoptions {
            file security-inteligence.log size 1g;
            level all;
            flag all;
        }
    }
```

Web Filtering (URL-Filterd) - Overview

The web filtering process reads the file contents fetched from the IPFD and configures the filters on the Packet Forwarding Engine accordingly. The web filtering process enforces the command and control feeds by programming the filters in the Packet Forwarding Engine to block the packets destined to the blocked IP addresses and to generate logs for reporting the incident.

Figure 5 illustrates the way C&C feed is fetched by the IPFD and then processed by the web filtering process.

Figure 52: Web Filtering


Juniper ATP Cloud
Threat Information
Junos Space Security Director
Policy Enforcer
Feed download from Juniper ATP Cloud is not targeted in first release
Command and Control Feed Download
Juniper Networks MX Series
ipfd (SecIntel)
C&C Feed
url-filterd
Program Filters
dfwd
Install Filters
PFE
No Match for Destination IP
IP Packets from Host
Implicit Filter for C&C Server IPs
Destination IP Matches
Drop Packet and Generate Log
g300602

The web filter profile can have more than one templates. Each template consists of a set of configured logical interfaces for Web filtering and one or more terms. A term is a set of match criteria with actions to be taken if the match criteria is met. To configure the web filter profile to use dynamically fetched C&C feed, you can configure the `security-intelligence-policy` command under the `[edit services web-filter profile` *`profile-name`* hierarchy level. You need not configure a term for a `security-intelligence-policy` based web filter profiles.

You can configure the following threat level actions for the web filter profile at the `edit web-filter profile` *`profile-name`* `security-intelligence-policy threat-level` *`threat-level`* `threat-action` hierarchy level:

- `drop`
- `drop-and-log`
- `log`

You can configure only one `threat-action` for each `threat level`. If the `threat-action` is not configured for a particular `threat level`, the default `threat-action` is `accept`.

SEE ALSO

security-intelligence-policy

security-intelligence

Configuring the Web Filter Profile for Sampling

IN THIS SECTION

- Associate a Sampling Instance with the FPC | 1117
- Configure a Sampling Instance and Associate the Template With the Sampling Instance. | 1118
- Configure the sample instance and associate the flow-server IP address and other parameters. | 1119
- Example: Configuring Web-filter Profile to Define Different Threat-Levels | 1120

Starting in Junos OS Release 19.3R1, web filtering process (url-filterd) supports inline sampling of packets as a threat level action. The packets are dropped, logged, and sampled based on the threat-action you configure. For scaled scenarios, sampling of packets is preferred over the logging option. Along with the existing threat level actions, you can configure the following threat level actions on the web filter profile at the `edit web-filter profile` *`profile-name`* `security-intelligence-policy threat-level` *`threat-level`* `threat-action` hierarchy level:

- `drop-and-sample`
- `drop-log-and-sample`
- `log-and-sample`
- `sample`

The inline flow monitoring samples the packets and sends the flow records in IPFIX format to a flow collector. You can derive the threat level for the sampled packets received at the external collector by matching the received IP from the sampled packets with the corresponding IP entry in **/var/db/url-filterd/urlf_si_cc_db.txt**. You can configure sampling using any of the following methods:

- Associate a sampling instance with the FPC on which the media interface is present at the `[edit chassis]` hierarchy level. If you are configuring sampling of IPv4 flows, IPv6 flows, or VPLS flows, you can configure the flow hash table size for each family.
- Configure the template properties for inline flow monitoring at the `[edit services flow-monitoring` hierarchy level.
- Configure a sampling instance and associate the flow-server IP address, port number, flow export rate, and specify the collectors at the `[edit forwarding-options` hierarchy level.

Associate a Sampling Instance with the FPC

To associate the defined instance with a particular FPC, MPC, or DPC, you include the `sampling-instance` statement at the `[edit chassis fpc number]` hierarchy level, as shown in the following example:

```
chassis {
    redundancy {
        graceful-switchover;
    }
    fpc 0 {
        pic0 {
            inline-services {
                bandwidth 10g;
            }
            }
        }
        pic 2 {
            inline-services {
                bandwidth 10g;
            }
        }
        pic 3 {
            inline-services {
                bandwidth 10g;
            }
        }
        sampling-instance 1to1;
        inline-services{
        flow-table-size {
             ipv4-flow-table-size 5;
             ipv6flow-table-size 5;
        }
    }
}
```

Configure a Sampling Instance and Associate the Template With the Sampling Instance.

To configure the template properties for inline flow monitoring, include the following statements at the `edit services flow-monitoring` hierarchy level as shown in the following example:

```
services {
    flow-monitoring {
        version-ipfix {
        template ipv4 {
        flow-active-timeout 60;
    flow-inactive-timeout 60;
        template-refresh-rate {
            packets 48000;
            seconds 60;
        }
        option-refresh-rate {
            packets 48000;
            seconds 60;
            }
        ipv4-template;
        template ipv6 {
            flow-active-timeout 60;
            flow-inactive-timeout 60;
            template-refresh-rate {
                packets 48000;
                seconds 60;
                }
                ipv6-template;
            }
        }
}
```

Configure the sample instance and associate the flow-server IP address and other parameters.

To configure a sampling instance and associate the flow-server IP address and other parameters. include the following statements at the `[edit forwarding-options]` hierarchy, as shown in the following example:

```
forwarding-options {
    sampling {
        traceoptions {
            file ipfix.log size 10k;
        }
    instance {
        1to1 {
        input {
        rate 1;
    }
    family inet {
        output {
            flow-server 192.168.9.194;
            port 2055;;
            autonomous-system-type origin;
            version-ipfix {
                template {
            ipv4;
        }
        }
        }
        inline-jflow {
        source-address 192.168.9.195;
        }
        }
    }
    family inet6 {
        output {
            flow-server 192.168.9.194;
            port 2000;
            autonomous-system-type origin;
            version-ipfix {
                template {
            ipv6;
        }
        }
        }
        inline-jflow {
        source-address  192.168.9.195;
        }
        }
    }
    }
}
```

Example: Configuring Web-filter Profile to Define Different Threat-Levels

```
web-filter {
    profile Profile1 ;
    security-intelligence-policy{
    file-type txt;
    threat-level 7 {
        threat-action {
        log-and-sample;
        }
    }
    threat-level 8 {
        threat-action {
        drop-log-and-sample;
        }
    }
    threat-level 10 {
        threat-action {
        drop-log-and-sample;
        }
    }
    threat-level 5{
        threat-action {
        drop-log-and-sample;
        }
    }
    threat-level 6 {
        threat-action {
        drop-log-and-sample;
        }
    }
    threat-level 9{
        threat-action {
        drop-log-and-sample;
        }
    }
    }
    url-filter-template template1 {
        client-interfaces ge-0/0/4.0;
        client-routing-instance inet.0;
    }
    }
    traceoptions {
        file webfilter_log size 1g;
        level all;
        flag all;
    }
    }
}
```

SEE ALSO

security-intelligence-policy

Configuring Traffic Sampling on MX, M and T Series Routers

GeoIP Filtering

IN THIS SECTION

- Overview | 1122
- How to Configure GeoIP Filtering on MX Series Routers | 1122

Overview

The GeoIP feeds are essentially a list of IP address to country code mappings. Starting in Junos OS 21.4R1, you can configure IP-based Geo locations on MX Series routers to fetch the GeoIP feeds from Policy Enforcer. By deploying the GeoIP feeds, you can enable the network to prevent devices from communicating with IP addresses belonging to specific countries.

You can configure the security intelligence process (IPFD) on MX series routers to fetch the GeoIP feeds from Policy Enforcer. Similar to existing C&C IP or IPv6 feeds, IPFD downloads the GeoIP feeds from the Policy Enforcer. IPFD translates the feed in the file format that is processed by the web-filtering process (url-filterd) subsequently.

Starting in Junos OS 22.1R1, you can configure the security intelligence process (IPFD) on MX series routers to fetch the GeoIP feeds from Juniper ATP Cloud. Similar to existing C&C IP or IPv6 feeds, IPFD downloads the GeoIP feeds from the Juniper ATP Cloud.

How to Configure GeoIP Filtering on MX Series Routers

The information fetched by the IPFD is saved in a file (**urlf_si_geoip_db.txt**) created at the **/var/db/url-filterd** location.

The format of the file sent by IPFD to the web filtering process is as follows:

`*IPv4 address*|*IPv6 address*,*Prefix*,*threat-level*,*VRF-name*,*Gen-num*`. Gen-num is always 0. *VRF-name* refers to a country code.

For example, 178.10.19.22,12,255,US,0

IPFD and the web-filtering process maintain a pconn connection for communicating the creation or update of files containing GeoIP feeds. The Web-Filtering process enforces the GeoIP feeds by programming the filters in the PFE to block the packets destined to the blocked countries. The APIs provided by liburlf are used to validate and parse the files.

The web-filtering process reads the file containing the list of IP addresses and the PFE filters are programmed with the destination IP addresses listed in the feed and the action configured for the associated country.

- Global filter- Countries are configured under global rule within a profile. All IP addresses for countries specific to that global rule are programmed in a single filter and applied to all templates in the profile. You can configure a profile to dynamically fetch GeoIP feed by configuring `geo-ip rule match country` *`country-name`* at the `[edit services web-filter profile` *`profile-name`* `security-intelligence-policy]` hierarchy .
- Group filter- Groups of countries are configured under a template. All IP addresses associated with the countries for a Group are programmed in a group filter applied to the templates under which that group is configured. Group is a list of countries defined in a json file that is parsed by liburlf.

To configure a group filter, you must configure a json file at the **/var/db/url-filterd** location, where the **group.json** file contains the group mappings.

The format of the json file is as follows:

```
[
{
"group_name" : "group1",
"country" : ["ZA","YE"]
},
{
"group_name" : "group2",
"country" : ["YT"]
}
]
```

To dynamically fetch GeoIP feeds, you can configure a global filter using a single profile or configure multiple group filters using templates. We do not support both the configurations together.

The groups created in the json file are referred in the GeoIP match clause defined at the `[edit services web-filter profile` *`profile-name`* `url-filter-template` *`template-name`* `security-intelligence-policy geo-ip rule match group` *`group-name`*`]` hierarchy.

Global Allowlist and Global Blocklist

You can choose to customize the IP feed by adding your own allowlist and blocklist. This can be helpful to manage intelligence feeds that are custom to your security operations center or as a temporary measure for false positives. Starting in Junos OS release 21.4R1, you can allow or block certain IP addresses based on configuration through a CLI or a file. You can either configure separate list for allowlist and a separate list for blocklist or include the IP addresses in a file and include the file name in the CLI configuration.

You can create an `IP-address-list` at the `[edit services web-filter]` hierarchy. Here, `IP-address-list` contains the list of IP addresses that must be allowed or blocked. You can also create a file containing the IP addresses that need to be allowed or blocked in the **/var/db/url-filterd** location. The IP addresses configured as a part of the file or IP address list are programmed as a part of the global filter, which is attached to all templates.

You can define a global allowlist by configuring `white-list (IP-address-list | file-name)` at the `edit services web-filter profile profile-name security-intelligence-policy` hierarchy. You can define a global blocklist by configuring the `black-list (IP-address-list | file-name)` at the `edit services web-filter profile profile-name security-intelligence-policy` hierarchy. Here, the *IP-address-list*, refers to the name of IP address-list specified at the `[edit services web-filter]` hierarchy. The *file-name* refers to the name of the file which contains the list of the IP addresses that must be allowed or blocked. The file must be in the **/var/db/url-filterd** location and must have the same name as in the configuration.

The format of the global allowlist file is as follows:

```
Security Intelligence Policy Enforcement Version 2.0
```

```
IP Address,Prefix,Threat-level,VRF-Name,Gen-Num
198.51.100.1,32,0,junos-default-vrf,0
```

The format of the global blocklist file is as follows:

```
Security Intelligence Policy Enforcement Version 2.0
```

```
IP Address,Prefix,Threat-level,VRF-Name,Gen-Num
192.168.1.1,255,junos-default-vrf,0
```

The web-filtering process parses the list of global allowlist or global blocklist IP addresses and programs the implicit filter terms with the configured IP addresses to either allow or block the packets.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
19.3R2	Starting in Junos OS Release 19.3R2, this same functionaly is supported for Next Gen Services on MX240, MX480, and MX960.
19.3R2	Starting in Junos OS Release 19.3R2, this same functionality is available for Next Gen Serices on MX240, MX480, and MX960.
19.3R2	Starting in Junos OS Release 19.3R2, you can configure DNS filtering if you are running Next Gen Services with the MX-SPC3 services card. Next Gen Services are supported on MX240, MX480 and MX960 routers.
19.3R2	Starting in Junos OS Release 19.3R2 with the Next Gen Services as an Inline security capability

19.3R1	Starting in Junos OS Release 19.3R1, web filtering process (url-filterd) supports inline sampling of packets as a threat level action
18.4R1	Starting in Junos OS Release 18.4R1 with the Adaptive Services as an Inline security capability
18.3R1	Starting in Junos OS Release 18.3R1, for Adaptive Services. configure the profile at the `[edit services web-filter]` hierarchy level. Before Junos OS Release 18.3R1, configure the profile at the `[edit services url-filter]` hierarchy level.
17.2R2	Starting in Junos OS Release 17.2R2 and 17.4R1, for Adaptive Services, you can disable the filtering of HTTP traffic that contains an embedded IP address (for example, http:/10.1.1.1) belonging to a disallowed domain name in the URL filter database.

16

PART

Configuration Statements and Operational Commands

unidirectional-session-refreshing | 1127

Junos CLI Reference Overview | 1128

unidirectional-session-refreshing

IN THIS SECTION

- Syntax | 1127
- Hierarchy Level | 1127
- Description | 1127
- Required Privilege Level | 1128
- Release Information | 1128

Syntax

```
unidirectional-session-refreshing  {
    input;
    output;
}
```

Hierarchy Level

```
[edit services service-set service-set-name service-set-options]
```

Description

For a service-set, enable unidirectional session refreshing in the forward direction (for the in-zone) and in the reverse direction (for the out-zone).

Required Privilege Level

interface—To view this statement in the configuration.

interface-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 21.2.

RELATED DOCUMENTATION

> No Link Title

Junos CLI Reference Overview

We've consolidated all Junos CLI commands and configuration statements in one place. Learn about the syntax and options that make up the statements and commands and understand the contexts in which you'll use these CLI elements in your network configurations and operations.

- Junos CLI Reference

Click the links to access Junos OS and Junos OS Evolved configuration statement and command summary topics.

- Configuration Statements
- Operational Commands