JUNIPER NETWORKS | Engineering Simplicity

Junos® OS

Broadband Subscriber Services User Guide

Published
2026-01-09

JUNOS

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

Juniper Networks, the Juniper Networks logo, Juniper, and Junos are registered trademarks of Juniper Networks, Inc. in the United States and other countries. All other trademarks, service marks, registered marks, or registered service marks are the property of their respective owners.

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 Broadband Subscriber Services User Guide
Copyright © 2026 Juniper Networks, Inc. All rights reserved.

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 | xviii

1 **Subscriber Service Activation and Management**

Subscriber Service Activation and Management | 2

Dynamic Service Management with RADIUS | 2

- Using RADIUS Dynamic Requests for Subscriber Access Management | 3
- Configuring RADIUS-Initiated Dynamic Request Support | 4
- RADIUS-Initiated Change of Authorization (CoA) Overview | 6
- RADIUS-Initiated Disconnect Overview | 10
- Usage Thresholds for Subscriber Services | 12
- Subscriber Session Logins and Service Activation Failures Overview | 13
- Configuring How Service Activation Failures Affect Subscriber Login | 18
- Error-Cause Codes (RADIUS Attribute 101) for Dynamic Requests | 19
- Verifying RADIUS Dynamic-Request Statistics | 20

Service Activation and Deactivation Using the CLI Instead of RADIUS | 21

- CLI-Activated Subscriber Services | 22
- Local and Remote Service Activation and Deactivation Using the CLI | 23

Management of Subscriber Services with Multiple Instances | 27

- Subscriber Services with Multiple Instances Overview | 27
- Deactivating a Single Instance of a Subscriber Service | 29
- Deactivating All Instances of a Subscriber Service | 33
- Verifying Subscriber Services with Multiple Instances | 36

Subscriber Management Redundancy for PFE During Graceful OIR | 38

Broadband Network Gateway (BNG) Subscriber Redundancy on Aggregated Ethernet (AE) Interfaces with Disabled PFEs (MX304, MX960, and MX10004) | 40

2 **Configuring Dynamic Class of Service**

CoS for Subscriber Access and Interfaces Overview | 43

CoS for Subscriber Access Overview | 43

Guidelines for Configuring Dynamic CoS for Subscriber Access | 44

CoS for Aggregated Ethernet Subscriber Interfaces Overview | 49

CoS for PPPoE Subscriber Interfaces Overview | 50

Configuring Scheduling and Shaping for Subscriber Access | 52

Configuring Traffic Scheduling and Shaping for Subscriber Access | 52

- Configuring Static Traffic Shaping and Scheduling Parameters in a Dynamic Profile | 53
- Configuring Dynamic Traffic Shaping and Scheduling Parameters in a Dynamic Profile | 54
- Using the CLI to Modify Traffic-Control Profiles That Are Currently Applied to Subscribers | 55
  - Using the CLI to Globally Modify a Traffic-Control Profile Currently Applied to Multiple Subscribers | 56
  - Using the CLI to Modify a Traffic-Control Profile for a Specific Current Subscriber | 57

Configuring Schedulers in a Dynamic Profile for Subscriber Access | 58

- Configuring Static Schedulers in a Dynamic Profile | 58
- Configuring Dynamic Schedulers with Variables in a Dynamic Profile | 60
- Configuring a Combination of Static and Dynamic Scheduler Parameters in a Scheduler Definition | 62

Configuring Scheduler and Scheduler Map Sharing | 66

Example: Providing Unique Rate Configurations for Schedulers in a Dynamic Profile | 68

Example: Configuring Aggregate Scheduling of Queues for Residential Subscribers on Static IP Demux Interfaces | 70

Verifying the Scheduling and Shaping Configuration for Subscriber Access | 72

Allocating Dedicated Queues for Each Logical Interface Using Per-Unit Scheduling | 74

Hardware Requirements for Dynamic Per-Unit Scheduling | 74

Configuring Per-Unit Scheduling in a Dynamic Profile | 75

Example: Configuring Per-Unit Scheduling for Subscriber Access | 77

Configuring Hierarchical CoS Scheduling on MPLS Ethernet Pseudowire Subscriber Interfaces | 89

Enhanced Subscriber Management Subscriber Logical Interfaces or Interface Sets Over Underlying Logical Interfaces for a CoS scheduler Hierarchy | 89

Enhanced Subscriber Management Subscriber Logical Interfaces or Interface Sets Over MPLS Pseudowires for a CoS scheduler Hierarchy | 92

Configuring Layer 2 Subscriber Logical Interfaces for CoS Hierarchical Schedulers Using Dynamic Profiles for Differentiating Home and Access Node Networks | **95**

Example: Configuring Layer 2 Subscriber Logical Interfaces for CoS Hierarchical Schedulers Using Static CoS for Differentiating Home and Access Node Networks | **101**

- Requirements | **101**
- Overview | **102**
- Configuration | **102**
- Verification | **105**

Configuring Dedicated Queue Scaling with Hierarchical CoS or Per-Unit Scheduling | 107

Dedicated Queue Scaling for CoS Configurations on MIC and MPC Interfaces Overview | **107**

Managing Dedicated and Remaining Queues for Dynamic CoS Configurations on MIC and MPC Interfaces | **110**

- Configuring the Maximum Number of Queues for MIC and MPC Interfaces | **111**
- Configuring Remaining Common Queues on MIC and MPC Interfaces | **111**

Verifying the Number of Dedicated Queues Configured on MIC and MPC Interfaces | **113**

Shaping Downstream Traffic Based on Frames or Cells | 115

Bandwidth Management for Downstream Traffic in Edge Networks Overview | **115**

Configuring Dynamic Shaping Parameters to Account for Overhead in Downstream Traffic Rates | **118**

Example: Configuring Dynamic Shaping Parameters to Account for Overhead in Downstream Traffic Rates | **119**

Configuring Static Shaping Parameters to Account for Overhead in Downstream Traffic Rates | **124**

Example: Configuring Static Shaping Parameters to Account for Overhead in Downstream Traffic Rates | **125**

Setting Shaping Rate and Overhead Accounting Based on PPPoE Vendor-Specific Tags | **128**

Configuring the Shaping Rate and Overhead Accounting Based on PPPoE Vendor-Specific Tags on Dynamic Subscriber Interfaces | **130**

Reporting the Effective Shaping Rate for Subscribers | **131**

Verifying the Effective Shaping Rate Reporting Configuration | **132**

Applying CoS to Households or Individual Subscribers Using ACI-Based Dynamic VLANs | 134

Applying CoS Attributes to VLANs Using Agent-Circuit-Identifiers | **134**

Agent Circuit Identifier-Based Dynamic VLANs Bandwidth Management Overview | **137**

Restrictions for Configuring Adjustment of CoS Shaping Rate and Overhead Accounting for Dynamic ACI Interface Sets | **141**

Adjusting the CoS Shaping Rate and Overhead Accounting Parameters for Agent Circuit Identifier-Based Dynamic VLANs | **142**

Applying CoS to Households or Individual Subscribers Using Access Line Identifier Dynamic VLANs | 144

Applying CoS Attributes to VLANs Using Access-Line Identifiers | **144**

Bandwidth Management Overview for Dynamic VLANs Based on Access-Line Identifiers | **147**

Restrictions for Configuring Adjustment of CoS Shaping Rate and Overhead Accounting for Dynamic ALI Interface Sets | **151**

Adjusting the CoS Shaping Rate and Overhead Accounting Parameters for Dynamic VLANs Based on Access-Line Identifiers | **152**

Managing Excess Bandwidth Distribution and Traffic Bursts | 154

Excess Bandwidth Distribution on MIC and MPC Interfaces Overview | **154**

Traffic Burst Management on MIC and MPC Interfaces Overview | **155**

Managing Excess Bandwidth Distribution for Dynamic CoS on MIC and MPC Interfaces | **158**

Applying CoS Using Parameters Received from RADIUS | 161

Subscriber Interfaces That Provide Initial CoS Parameters Dynamically Obtained from RADIUS | **161**

Changing CoS Services Overview | **166**

CoS Traffic Shaping Attributes for Dynamic Interface Sets and Member Subscriber Sessions Overview | **170**

Guidelines for Configuring CoS Traffic Shaping Attributes for Dynamic Interface Sets and Member Subscriber Sessions | **172**

Configuring Initial CoS Parameters Dynamically Obtained from RADIUS | **173**

Configuring Static Default Values for Traffic Scheduling and Shaping | **174**

Applying CoS Traffic-Shaping Attributes to Dynamic Interface Sets and Member Subscriber Sessions | **176**

CoS Traffic Shaping Predefined Variables for Dynamic Interface Sets | **179**

Example: Configuring Initial CoS Parameters Dynamically Obtained from RADIUS | **185**

Modifying a Subscriber' s Shaping Characteristics After a Subscriber is Instantiated | 189

CoS Adjustment Control Profiles Overview | **189**

Configuring CoS Adjustment Control Profiles | **192**

Verifying the CoS Adjustment Control Profile Configuration | **194**

Applying CoS to Groups of Subscriber Interfaces | 196

CoS for Interface Sets of Subscribers Overview | **196**

Configuring an Interface Set of Subscribers in a Dynamic Profile | **199**

Example: Configuring a Dynamic Interface Set of VLAN Subscribers | **200**

- Requirements | **200**
- Overview | **200**
- Configuring the Dynamic VLANs | **201**
- Configuring Dynamic Traffic Scheduling and Shaping | **204**
- Configuring the Interface Set in the Dynamic Profile | **209**
- Configuring DHCP Access | **212**
- Configuring RADIUS Authentication | **214**
- Verification | **220**

Example: Configure a Dynamic Service VLAN Interface Set of Subscribers in a Dynamic Profile | **221**

- Requirements | **222**
- Overview | **222**
- Configuration | **223**
- Verification | **226**

Applying CoS to Subscriber Interfaces | 228

Applying Traffic Shaping and Scheduling to a Subscriber Interface in a Dynamic Profile | **228**

Applying Minimal Shaping and Scheduling to Remaining Subscriber Traffic | **230**

Applying a Rewrite Rule Definition to a Subscriber Interface in a Dynamic Profile | **231**

Applying a Classifier to a Subscriber Interface in a Dynamic Profile | **232**

3 **Configuring Dynamic Filters and Policers**

Dynamic Firewall Filters Overview | 236

Understanding Dynamic Firewall Filters | **236**

Defining Dynamic Filter Processing Order | **237**

Configuring Static Firewall Filters That Are Dynamically Applied | 239

Classic Filters Overview | **239**

Basic Classic Filter Syntax | **243**

Examples: Configuring Static Filters | **244**

Streamlining Processing of Chains of Static Filters | 248

Configuring Firewall Filter Bypass | **248**

Example: Bypassing Firewall Filters | **249**

- Before You Begin | **250**
- Filter Bypass Overview | **250**
- Configuring Filter Bypass | **250**

Dynamically Attaching Static or Fast Update Filters to an Interface | 255

Dynamically Attaching Statically Created Filters for a Specific Interface Family Type | **255**

Dynamically Attaching Statically Created Filters for Any Interface Type | **256**

Configuring Filters That Are Created Dynamically | 258

Parameterized Filters Overview | **258**

Unique Identifiers for Firewall Variables | **259**

Configuring Unique Identifiers for Parameterized Filters | **262**

Sample Dynamic-Profile Configuration for Parameterized Filters | **263**

Dynamic Profile After UID Substitutions for Parameterized Filters | **266**

Multiple Parameterized Filters | **268**

Parameterized Filter Processing Overview | **268**

Parameterized Filters Configuration Considerations | **270**

Guidelines for Creating and Applying Parameterized Filters for Subscriber Interfaces | **271**

Parameterized Filter Match Conditions for IPv4 Traffic | **272**

Parameterized Filter Match Conditions for IPv6 Traffic | **281**

Parameterized Filter Nonterminating and Terminating Actions and Modifiers | **291**

Firewall Filter Match Conditions for Protocol-Independent Traffic in Dynamic Service Profiles | **300**

Firewall Filter Terminating and Nonterminating Actions for Protocol-Independent Traffic in Dynamic Service Profiles | **302**

Interface-Shared Filters Overview | **306**

Dynamically Attaching Filters Using RADIUS Variables | **307**

Example: Implementing a Filter for Households That Use ACI-Based VLANs | **309**

Example: Dynamic-Profile Parsing | **312**

Example: Firewall Dynamic Profile | **313**

Example: Configuring a Filter to Exclude DHCPv6 and ICMPv6 Control Traffic for LAC Subscriber | **314**

- Requirements | **314**
- Overview | **314**
- Configuration | **315**

Using Ascend Data Filters to Implement Firewalls Based on RADIUS Attributes | 321

Ascend-Data-Filter Policies for Subscriber Management Overview | **321**

Ascend-Data-Filter Attribute Fields | **324**

Dynamically Applying Ascend-Data-Filter Policies to Subscriber Sessions | **328**

Example: Configuring Dynamic Ascend-Data-Filter Support for Subscriber Access | **331**

- Requirements | **331**
- Overview | **331**
- Configuration | **332**
- Verification | **334**

Example: Configuring Static Ascend-Data-Filter Support for Subscriber Access | **336**

- Requirements | **337**
- Overview | **337**
- Configuration | **337**
- Verification | **340**

Verifying and Managing Dynamic Ascend-Data-Filter Policy Configuration | **342**

Configuring Fast Update Filters to Provide More Efficient Processing Over Classic Static Filters | 344

Fast Update Filters Overview | **344**

Basic Fast Update Filter Syntax | **348**

Configuring Fast Update Filters | 349

Example: Configuring Fast Update Filters for Subscriber Access | 351

Match Conditions and Actions in Fast Update Filters | 352

Configuring the Match Order for Fast Update Filters | 354

Fast Update Filter Match Conditions | 355

Fast Update Filter Actions and Action Modifiers | 356

Configuring Terms for Fast Update Filters | 357

Configuring Filters to Permit Expected Traffic | 359

Avoiding Conflicts When Terms Match | 360

Associating Fast Update Filters with Interfaces in a Dynamic Profile | 367

Defending Against DoS and DDoS Attacks Using Unicast RPF and Fail Filters | 369

Unicast RPF in Dynamic Profiles for Subscriber Interfaces | 369

- Unicast RPF in Dynamic Profiles for Subscriber Interfaces | 369
- Configuring Unicast RPF in Dynamic Profiles for Subscriber Interfaces | 370
- Configuring a Fail Filter for Unicast RPF in Dynamic Profiles for Subscriber Interfaces | 371
- Example: Configuring Unicast RPF in a Dynamic Profile on MX Series Routers | 372
  - Requirements | 373
  - Overview | 373
  - Configuration | 374
  - Verification | 379

Improving Scaling and Performance of Filters on Static Subscriber Interfaces | 381

Firewall Filters and Enhanced Network Services Mode Overview | 381

Configuring a Filter for Use with Enhanced Network Services Mode | 384

Configuring Dynamic Service Sets | 386

Dynamic Service Sets Overview | 386

Associating Service Sets with Interfaces in a Dynamic Profile | 387

Verifying and Managing Service Sets Information | 388

Configuring Rate-Limiting Premium and Non-Premium Traffic on an Interface Using Hierarchical Policers | 390

Methods for Regulating Traffic by Applying Hierarchical Policers | **390**

Hierarchical Policer Applied as Filter Action | **393**

Example: Configuring Hierarchical Policers to Limit Rates of Services in a Static Environment | **394**

- Requirements | **395**
- Overview | **395**
- Configuration | **397**
- Verification | **407**

Configure Rate-Limiting Traffic on an Interface Using Enhanced Hierarchical Policers | 412

Enhanced Hierarchical Policer Overview (Junos OS Evolved) | **412**

Configure Enhanced Hierarchical Policer (Junos OS Evolved) | **414**

Verify Enhanced Hierarchical Policer Configuration | **423**

Monitoring and Managing Firewalls for Subscriber Access | 426

Verifying and Managing Firewall Filter Configuration | **426**

Enhanced Policer Statistics Overview | **427**

4 **Configuring Dynamic Multicast**

Configuring Dynamic IGMP to Support IP Multicasting for Subscribers | 430

Dynamic IGMP Configuration Overview | **430**

Subscriber Management IGMP Model Overview | **431**

Configuring Dynamic DHCP Client Access to a Multicast Network | **432**

Example: IGMP Dynamic Profile | **434**

Configuring SSM Mapping for Dynamic IGMP and MLD | **436**

Configuring Dynamic MLD to Enable Subscribers to Access Multicast Networks | 439

Dynamic MLD Configuration Overview | **439**

5 **Configuring Application-Aware Policy Control and Reporting**

Configuring Application-Aware Policy Control | 442

Understanding Application-Aware Policy Control for Subscriber Management | **442**

Understanding PCC Rules for Subscriber Management | **444**

Configuring Application-Aware Policy Control for Subscriber Management | **446**

Installing Services Packages for Subscriber Management Application-Aware Policy Management | 447

Configuring Service Data Flow Filters | 448

Configuring Policy and Charging Control Action Profiles for Subscriber Management | 452

Configuring Policy and Charging Control Rules | 454

Configuring a Policy and Charging Control Rulebase | 458

Configuring a Policy and Charging Enforcement Function Profile for Subscriber Management | 460

Identifying the Service Interface That Handles Subscriber Management Application-Aware Policy Control | 462

Configuring PCC Rule Activation in a Subscriber Management Dynamic Profile | 463

Enabling Direct PCC Rule Activation by a PCRF for Subscriber Management | 465

Configuring Application Identification | 468

Application Identification Overview | 468

Downloading and Installing Predefined Junos OS Application Signature Packages | 470

Improving the Application Traffic Throughput | 471

Configuring Custom Application Signatures | 472

Uninstalling a Predefined Junos OS Application Signature Package | 478

Configuring Reporting for Application-Aware Data Sessions | 479

Logging and Reporting Function for Subscribers | 479

Log Dictionary for Template Types | 487

Configuring Logging and Reporting for Subscriber Management | 497

Installing Services Packages for Subscriber Management Logging and Reporting | 498

Configuring an LRF Profile for Subscribers | 499

- Configuring the LRF Profile Name | 499
- Configuring Policy-Based Logging | 500
- (Optional) Configuring HTTP Transaction Logging | 500
- Configuring Collectors | 500
- Configuring Templates | 502
- Configuring Logging and Reporting Rules | 503

Applying Logging and Reporting Configuration to a Subscriber Management Service Set | **505**

Configuring the Activation of an LRF Rule by a PCC Rule | **506**

6 **Configuring HTTP Redirect Services**

Configuring Captive Portal Content Delivery Services for Redirected Subscribers | 511

HTTP Redirect Service Overview | **511**

Remote HTTP Redirect Server Operation Flow | **518**

Local HTTP Redirect Server Operation Flow (MX Series, ACX7100-48L, ACX7332 and ACX7348) | **520**

Configuring MS-MPC-Based or MX-SPC3-Based Static HTTP Redirect Services | **522**

- Configuring a Walled Garden as a Firewall Service Filter | **523**
- Configuring HTTP Redirect for Local and Remote Redirect Servers | **527**
- Configuring the Service Profile and the Service Set to Associate the Service Profile with a Service Interface | **528**
- Attaching a CPCD Service Set and Service Filter to a Logical Interface | **530**
- Installing a Service Package for CPCD Service | **531**

Configuring MS-MPC-Based or MX-SPC3-Based Converged HTTP Redirect Services | **532**

- Configuring a Walled Garden as a Firewall Service Filter | **533**
- Configuring HTTP Redirect for Local and Remote Redirect Servers | **537**
- Configuring Parameterization for the Redirect URL | **538**
- Configuring the Service Set to Associate the Service Profile with a Service Interface | **540**
- Attaching a CPCD Service Set and Service Filter to a Dynamic Logical Interface | **541**
- Installing a Service Package for CPCD Service | **543**

Configuring Routing Engine-Based, Static HTTP Redirect Services | **544**

- Configuring a Walled Garden as a Firewall Service Filter | **545**
- Configuring HTTP Redirect for Local and Remote Redirect Servers | **549**
- Configuring the Service Profile and the Service Set to Associate the Service Profile with a Service Interface | **551**
- Attaching a CPCD Service Set and Service Filter to a Logical Interface | **552**
- Inserting GET Header Tags That the HTTP Server Can Use to Control Content Access | **553**

Configuring Routing Engine-Based, Converged HTTP Redirect Services | **559**

- Configuring a Walled Garden as a Firewall Service Filter | **560**
- Configuring HTTP Redirect for Local and Remote Redirect Servers | **563**
- Configuring Parameterization for the Redirect URL | **565**

Configuring the Service Set to Associate the Service Profile with a Service Interface | 567

Attaching a CPCD Service Set and Service Filter to a Dynamic Logical Interface | 568

Adding Subscriber Information to HTTP Redirect URLs | 570

How to Automatically Remove the HTTP Redirect Service After the Initial Redirect | 572

Example: Configuring HTTP Redirect Services Using a Next-Hop Method and Attaching It to a Static Interface | 575

Requirements | 575

Overview | 575

Configuration | 576

Verification | 592

7 Configuring Subscriber Secure Policy

Configuring Subscriber Secure Policy Traffic Mirroring Overview | 595

Subscriber Secure Policy Overview | 595

Configuring RADIUS-Initiated Subscriber Secure Policy Traffic Mirroring | 600

RADIUS-Initiated Subscriber Secure Policy Overview | 600

Subscriber Secure Policy Traffic Mirroring Architecture Using RADIUS | 601

RADIUS-Initiated Traffic Mirroring Interfaces | 603

RADIUS-Initiated Traffic Mirroring Process at Subscriber Login | 605

RADIUS-Initiated Traffic Mirroring Process for Logged-In Subscribers | 607

RADIUS Attributes Used for Subscriber Secure Policy | 608

Using the Packet Header to Track Subscribers on the Mediation Device | 610

Configuring RADIUS-Initiated Subscriber Secure Policy Mirroring Overview | 616

Guidelines for Configuring Subscriber Secure Policy Mirroring | 617

Configuring Support for Subscriber Secure Policy Mirroring | 619

Configuring RADIUS Server Support for Subscriber Secure Policy Mirroring | 622

Terminating RADIUS-Initiated Subscriber Traffic Mirroring | 623

Configuring DTCP-Initiated Subscriber Secure Policy Traffic Mirroring | 624

DTCP-Initiated Subscriber Secure Policy Overview | 624

Subscriber Secure Policy Traffic Mirroring Architecture Using DTCP | **625**

DTCP-Initiated Traffic Mirroring Interfaces | **627**

DTCP-Initiated Traffic Mirroring Process | **628**

DTCP Messages Used for Subscriber Secure Policy | **630**

Packet Header for Mirrored Traffic Sent to Mediation Device | **631**

Configuring DTCP-Initiated Subscriber Secure Policy Mirroring Overview | **636**

Guidelines for Configuring Subscriber Secure Policy Mirroring | **637**

Configuring Support for Subscriber Secure Policy Mirroring | **638**

Configuring the Mediation Device as a User on the Router | **641**

Configuring a DTCP-over-SSH Connection to the Mediation Device | **643**

Configuring the Mediation Device to Provision Traffic Mirroring | **644**

Disabling RADIUS-Initiated Subscriber Secure Policy Mirroring | **644**

Example: Configuring Traffic That Is Mirrored Using DTCP-Initiated Subscriber Secure Policy | **645**

- Requirements | **645**
- Overview | **645**
- Configuration | **646**

Terminating DTCP-Initiated Subscriber Traffic Mirroring Sessions | **648**

Configuring DTCP Messages Used for DTCP-Initiated Subscriber Secure Policy Mirroring | 649

ADD (DTCP) | **649**

DELETE (DTCP) | **655**

DISABLE (DTCP) | **657**

ENABLE (DTCP) | **659**

LIST (DTCP) | **662**

Example: Using DTCP Messages to Trigger, Verify, and Remove Traffic Mirroring for Subscribers | **665**

Configuring Subscriber Secure Policy Support for IPv4 Multicast Traffic | 672

Subscriber Secure Policy Support for IPv4 Multicast Traffic | **672**

Enabling Subscriber Secure Policy Mirroring for IPv4 Multicast Traffic | **673**

Configuring Intercept-Related Information for Subscriber Secure Policy | 675

Intercept-Related Events Transmitted to the Mediation Device | **675**

SNMP Traps for Subscriber Secure Policy LAES Compliance | **676**

Configuring SNMPv3 Traps for Subscriber Secure Policy Mirroring | **678**

Example: SNMPv3 Traps Configuration for Subscriber Secure Policy Mirroring | **679**

8 **Configuring Stateless, Rule-Based Services Using Application-Aware Access Lists**

AACL Overview | 682

AACL Overview | **682**

Configuring AACL Rules | 684

Configuring AACL Rules | **684**

Example: Configuring AACL Rules | 690

Example: Configuring AACL Rules | **690**

Example: Configuring AACL Rule Sets | 692

Configuring AACL Rule Sets | **692**

Configuring Logging of AACL Flows | 693

Configuring Logging of AACL Flows | **693**

9 **Remote Device and Service Management**

Configuring Remote Device Services Management | 696

Remote Device Services Manager (RDSM) Overview | **696**

Configuring Remote Device Management for Service Provisioning | **715**

Reconfiguring a Remote Device for RDSM | **720**

Reloading a Dictionary File for RDSM | **721**

Configuring TCP Port Forwarding for Remote Subscriber Services | 722

TCP Port Forwarding for Remote Device Management | **722**

Configure TCP Port Forwarding for Remote Device Management | **725**

Tracing TCP Port Forwarding Events for Troubleshooting | **729**

- Configuring the TCP Port Forwarding Trace Log Filename | **730**
- Configuring the Number and Size of TCP Port Forwarding Log Files | **730**
- Configuring Access to the TCP Port Forwarding Log File | **730**
- Configuring a Regular Expression for TCP Port Forwarding Messages to Be Logged | **731**
- Configuring the TCP Port Forwarding Tracing Flags | **731**
- Configuring the Severity Level to Filter Which TCP Port Forwarding Messages Are Logged | **732**

Configuring IPFIX Mediation for Remote Device Monitoring | 733

IPFIX Mediation on the BNG | **733**

Configuring the BNG as an IPFIX Mediator to Collect and Export IPFIX Data | **739**

Collection and Export of Local Telemetry Data on the IPFIX Mediator | 743

Telemetry Data Collection on the IPFIX Mediator for Export to an IPFIX Collector | **743**

Configuring the Collection and Export of Local Telemetry Data on the IPFIX Mediator | **747**

10 **Troubleshooting**

Contacting Juniper Networks Technical Support | 753

Collecting Subscriber Access Logs Before Contacting Juniper Networks Technical Support | **753**

Knowledge Base | 756

11 **Configuration Statements and Operational Commands**

Junos CLI Reference Overview | 758

About This Guide

Use this guide to understand conceptual and configuration information about dynamic class of service, policy filters, and traffic policing; dynamic IGMP and MLD for access to multicast networks; application-aware policy control; HTTP redirect services to capture subscriber network requests and send them to a captive portal for authentication and access to authorized Web resources; and subscriber secure policy traffic mirroring to mirror subscriber traffic and monitor events related to the mirrored session.

1

PART

Subscriber Service Activation and Management

- Subscriber Service Activation and Management | 2

CHAPTER 1

Subscriber Service Activation and Management

IN THIS CHAPTER

- Dynamic Service Management with RADIUS | **2**
- Service Activation and Deactivation Using the CLI Instead of RADIUS | **21**
- Management of Subscriber Services with Multiple Instances | **27**
- **Subscriber Management Redundancy for PFE During Graceful OIR** | **38**
- Broadband Network Gateway (BNG) Subscriber Redundancy on Aggregated Ethernet (AE) Interfaces with Disabled PFEs (MX304, MX960, and MX10004) | **40**

Dynamic Service Management with RADIUS

IN THIS SECTION

- Using RADIUS Dynamic Requests for Subscriber Access Management | **3**
- Configuring RADIUS-Initiated Dynamic Request Support | **4**
- RADIUS-Initiated Change of Authorization (CoA) Overview | **6**
- RADIUS-Initiated Disconnect Overview | **10**
- Usage Thresholds for Subscriber Services | **12**
- Subscriber Session Logins and Service Activation Failures Overview | **13**
- Configuring How Service Activation Failures Affect Subscriber Login | **18**
- Error-Cause Codes (RADIUS Attribute 101) for Dynamic Requests | **19**
- Verifying RADIUS Dynamic-Request Statistics | **20**

Using RADIUS Dynamic Requests for Subscriber Access Management

IN THIS SECTION

- Benefits of Radius Dynamic Requests | 4

RADIUS dynamic requests provide an efficient way to centrally manage subscriber sessions. The AAA Service Framework's RADIUS dynamic request support allows RADIUS servers to initiate user-related operations, such as a termination operation, by sending unsolicited request messages to the router. Without the RADIUS dynamic request feature, the only way to disconnect a RADIUS user is from the router, which can be cumbersome and time-consuming in large networks.

In a typical client-server RADIUS environment, the router functions as the client and initiates requests sent to the remote RADIUS server. However, when using RADIUS dynamic requests, the roles are reversed. For example, during a disconnect operation, the remote RADIUS server performs as the client and initiates the request (the disconnect action) — the router functions as the server in the relationship.

You create an access profile to configure the router to support RADIUS dynamic requests. This configuration enables the router to receive and act on the following types of messages from remote RADIUS servers:

- Access-Accept messages—Dynamically activate services based on attributes in RADIUS Access-Accept messages received when a subscriber logs in.
- Change-of-Authorization (CoA) messages—Dynamically modify active sessions based on attributes in CoA messages. CoA messages can include service creation requests, deletion requests, RADIUS attributes, and Juniper Networks VSAs.
- Disconnect messages—Immediately terminate specific subscriber sessions.

By default, the router monitors UDP port 3799 for CoA requests from RADIUS servers. You can also configure a nondefault port for RADIUS servers. You must either use the default port for all RADIUS servers or configure the same nondefault port for all RADIUS servers. This rule applies at both the global access and access profile levels.

NOTE: Any other configuration results in a commit check failure. Multiple port numbers —that is, different port numbers for different servers—are not supported.

Benefits of Radius Dynamic Requests

Enables simplified central management of subscriber sessions by sending unsolicited changes to subscriber sessions, including changes in attributes, service activation, service deactivation, and session termination.

SEE ALSO

RADIUS-Initiated Change of Authorization (CoA) Overview | 6

RADIUS-Initiated Disconnect Overview | 10

Configuring RADIUS-Initiated Dynamic Request Support | 4

RADIUS Attributes and Juniper Networks VSAs Supported by the AAA Service Framework

Error-Cause Codes (RADIUS Attribute 101) for Dynamic Requests | 19

Configuring RADIUS-Initiated Dynamic Request Support

The router uses the list of specified RADIUS authentication servers for both authentication and dynamic request operations. By default, the router monitors UDP port 3799 for dynamic requests, also known as Change of Authorization (CoA) requests.

To configure RADIUS dynamic request support:

- Specify the IP address of the RADIUS server.

```
[edit access profile isp-bos-metro-fiber-basic radius]
user@host# set authentication-server 192.168.1.3
```

To configure the router to support dynamic requests from more than one RADIUS server:

- Specify the IP addresses of multiple RADIUS servers.

```
[edit access profile isp-bos-metro-fiber-basic radius]
user@host# set authentication-server 192.168.1.3 192.168.10.15
```

When you configure dynamic request ports, you must do one of the following:

- Use the default port for all RADIUS servers at both the global access level and in all access profiles.
- Configure the same nondefault port for all servers at both the global access level and in all access profiles.

NOTE: Any other configuration results in a commit check failure. Multiple port numbers —that is, different port numbers for different servers—are not supported.

To specify a global dynamic request port:

```
[edit access]
user@host# set radius-server server-address dynamic-request-port port-number
```

To specify the dynamic request port for a specific access profile:

```
[edit access]
user@host# set profile profile-name radius-server server-address dynamic-request-port port-number
```

Consider the following scenarios:

- The following configuration uses the default port for both a server globally and a different server in the access profile. This is a valid configuration.

  ```
  [edit access]
  user@host# set radius-server 192.0.2.1
  user@host# set profile ap1 radius-server 192.0.2.3
  ```

- The following configuration specifies nondefault port 50201 for both a server globally and a different server in the access profile. This is a valid configuration.

  ```
  [edit access]
  user@host# set radius-server 192.0.2.1 dynamic-request-port 50201
  user@host# set profile ap1 radius-server 192.0.2.3 dynamic-request-port 50201
  ```

- The following configuration specifies port 50201 globally for a server and port 51133 for the same server in the ap1 access profile. This is an invalid configuration and commit check fails, because multiple nondefault ports are not supported.

  ```
  [edit access]
  user@host# set radius-server 192.0.2.1 dynamic-request-port 50201
  user@host# set profile ap1 radius-server 192.0.2.1 dynamic-request-port 51133
  ```

- The following configuration uses the default port 3799 for one server globally and port 51133 for another server globally. This is an invalid configuration and the commit check fails, because for all servers you must configure either the default port or the same nondefault port.

```
[edit access]
user@host# set radius-server 192.0.2.1
user@host# set radius-server 192.0.2.3 dynamic-request-port 51133
```

SEE ALSO

RADIUS Attributes and Juniper Networks VSAs Supported by the AAA Service Framework

RADIUS-Initiated Change of Authorization (CoA) Overview

IN THIS SECTION

- CoA Messages | 6
- Qualifications for Change of Authorization | 7
- Message Exchange | 8
- Bulk CoA Transactions | 9
- Benefits of Radius-Initiated Change of Authorization | 10

The AAA Service Framework uses CoA messages to dynamically modify active subscriber sessions. For example, RADIUS attributes in CoA messages might instruct the framework to create, modify, or terminate a subscriber service. You can also use CoA messages to set or modify usage thresholds for current subscriber services.

CoA Messages

Dynamic request support enables the router to receive and process unsolicited CoA messages from external RADIUS servers. RADIUS-initiated CoA messages use the following codes in request and response messages:

- CoA-Request (43)
- CoA-ACK (44)

- CoA-NAK (45)

Qualifications for Change of Authorization

To complete the change of authorization for a user, you specify identification attributes and session attributes. The identification attributes identify the subscriber. Session attributes specify the operation (activation or deactivation) to perform on the subscriber's session and also include any client attributes for the session (for example, QoS attributes). The AAA Service Framework handles the actual request.

Table 1 on page 7 shows the identification attributes for CoA operations.

Table 1: Identification Attributes

Attribute	Description
User-Name [RADIUS attribute 1]	Subscriber username.
Acct-Session-ID [RADIUS attribute 44]	Specific subscriber session.

NOTE: Using the Acct-Session-ID attribute to identify the subscriber session is more explicit than using the User-Name attribute. When you use the User-Name as the identifier, the CoA operation is applied to the first session that was logged in with the specified username. However, because a subscriber might have multiple sessions associated with the same username, the first session might not be the correct session for the CoA operation.

When you use the Acct-Session-ID attribute, it identifies the specific subscriber session, avoiding that potential error. Although the Acct-Session-ID attribute can include an interface specifier in addition to the session ID—when the attribute is in the description format—only the session ID is used for subscriber matching. For example, if the subscriber has a subscriber session ID of `54785`, then the subscriber is matched when the Acct-Session-ID attribute is `54785` (decimal format), or `jnpr demux0.1073759682:54785` (description format), or indeed *`any value`*`:54785` (description format).

Table 2 on page 8 shows the session attributes for CoA operations. Any additional client attributes that you include depend on your particular session requirements.

Table 2: Session Attributes

Attribute	Description
Activate-Service [Juniper Networks VSA 26-65]	Service to activate for the subscriber.
Deactivate-Service [Juniper Networks VSA 26-66]	Service to deactivate for the subscriber.
Service-Volume [Juniper Networks VSA 26-67]	Amount of traffic, in MB, that can use the service; service is deactivated when the volume is exceeded.
Service-Timeout [Juniper Networks VSA 26-68]	Number of seconds that the service can be active; service is deactivated when the timeout expires.
Service-Volume-Gigawords [Juniper Networks VSA 26-179]	Amount of traffic, in 4GB units, that can use the service; service is deactivated when the volume is exceeded.
Update-Service [Juniper Networks VSA 26-180]	New values of service and time quotas for existing service.

Message Exchange

The RADIUS server and the AAA Service Framework on the router exchange messages using UDP. The CoA-Request message sent by the RADIUS server has the same format as the Disconnect-Request packet that is sent for a disconnect operation.

The response is either a CoA-ACK or a CoA-NAK message:

- If the AAA Service Framework successfully changes the authorization, the response is a RADIUS-formatted packet with a CoA-ACK message, and the data filter is applied to the session.
- If AAA Service Framework is unsuccessful, the request is malformed, or attributes are missing, the response is a RADIUS-formatted packet with a CoA-NAK message.

NOTE: The AAA Service Framework processes one dynamic request at a time per subscriber. If the framework receives a second dynamic request (either another CoA or a Disconnect-Request) while processing a previous request for the same subscriber, the

framework responds with a CoA-NAK message. Starting in Junos OS Release 15.1R5, CoA-Request retry messages are ignored and no CoA-NAK is sent in response to them.

Bulk CoA Transactions

Starting in Junos OS Release 17.2R1, bulk CoA requests are supported to improve the processing efficiency of RADIUS-based subscriber services on the BNG. The bulk CoA functionality enables the accumulation of a series of CoA requests and commits all of them together, in bulk, automatically.

All the services in a bulk CoA request must be for the same subscriber session.

Bulk CoA transactions are particularly valuable for business services. RADIUS-based subscriber services use the Juniper Networks VSAs, Activate-Service (26-65) and Deactivate-Service (26-66). The VSAs are provided in RADIUS-Accept messages during login or in CoA requests after login.

For conventional, dynamic service profile-based services, up to 12 service activations can easily fit within either RADIUS message. However, the op-script based services used by businesses typically have scaling requirements that exceed the capacity of either message. This means that specifying and activating all the services needed in a given subscriber session may require using an Accept-Access message and multiple CoA requests.

Each CoA request message is independent of previous and future CoA requests in the same subscriber session. All service-activations and deactivations in a message are processed before a CoA response is offered. The CoA request provides a way to incrementally modify a subscriber session without affecting existing services that are already present.

For op-script based services, the service sessions are collaboratively created by the authd and essmd processes such that the last operation initiates a commit to apply all resultant static business logical interfaces from the CoA request. Because the commit time is generally the largest part of applying a static business service, there is an advantage to packing as many service-activations or deactivations as will fit within a RADIUS message to efficiently use the commit window. Until the commit operation completes, the BNG cannot accept a subsequent CoA request to apply additional business services for the same subscriber session.

Bulk CoA improves the efficiency of commit processing by using a single commit action for all services in the bulk transaction. The bulk transaction has the effect of managing a series of requests as a single meta-request. It defers the commit processing until the final CoA request in the bulk transaction is received.

Bulk CoA requires each individual request to contain a single instance of the Juniper Networks Bulk-CoA-Transaction-Id VSA (26–194). This VSA identifies requests as belonging to the same bulk transaction; 26–194 must have the same value in all CoA requests in the bulk series. Each successive bulk transaction in the session must have a different identifier; for example, three successive bulk transactions can have IDs of 1, 2, and 1, but cannot have successive IDs of 1, 1, and 2. In practice, the

Bulk-CoA-Transaction-Id value typically is incremented for multiple bulk transactions, but this is not required. An ID value used in a given subscriber session can also be used in different subscriber sessions.

Each CoA request within a bulk transaction has its own unique identifier, provided by a single instance of the Bulk-CoA-Identifier VSA (26–195) in each CoA. An increasing series of values for the ID is typical but not enforced. Values can be reused within a given session and between sessions. The final CoA request in the series is identified by having a value of 0xFFFFFFFF for the Bulk-CoA-Identifier.

Benefits of Radius-Initiated Change of Authorization

Enables changes in attribute values to be dynamically pushed to subscriber sessions, as well as dynamic activation and deactivation of subscriber services.

SEE ALSO

- Using RADIUS Dynamic Requests for Subscriber Access Management | 3
- RADIUS-Initiated Disconnect Overview | 10
- Configuring RADIUS-Initiated Dynamic Request Support | 4
- Usage Thresholds for Subscriber Services | 12
- *RADIUS Attributes and Juniper Networks VSAs Supported by the AAA Service Framework*

RADIUS-Initiated Disconnect Overview

IN THIS SECTION

- Disconnect Messages | 11
- Qualifications for Disconnect | 11
- Message Exchange | 11
- Benefits of Radius-Initiated Disconnects | 12

This section describes the AAA Service Framework's support for RADIUS-initiated disconnect dynamic requests. The AAA Service Framework uses disconnect messages to dynamically terminate active subscriber sessions.

Disconnect Messages

To centrally control the disconnection of remote access subscribers, the RADIUS dynamic request feature on the router receives and processes unsolicited messages from RADIUS servers.

The dynamic request feature uses the existing format of RADIUS disconnect request and response messages. RADIUS-initiated disconnect uses the following codes in its RADIUS request and response messages:

- Disconnect-Request (40)
- Disconnect-ACK (41)
- Disconnect-NAK (42)

Qualifications for Disconnect

For the AAA Service Framework to disconnect a user, the Disconnect-Request message must contain an attribute with an accounting session ID. The Disconnect-Request message can contain an Acct-Session-Id (44) attribute or an Acct-Multi-Session-Id (50) attribute for the session ID or both. If both the Acct-Session-Id and Acct-Multi-Session-Id attributes are present in the request, the router uses both attributes. If the User-Name (1) attribute is also present in the request, the username and accounting session ID are used to perform the disconnection. The AAA Service Framework handles the actual request.

Message Exchange

The RADIUS server and the AAA Service Framework exchange messages using UDP. The Disconnect-Request message sent by the RADIUS server has the same format as the CoA-Request packet that is sent for a change of authorization operation.

The disconnect response is either a Disconnect-ACK or a Disconnect-NAK message:

- If the AAA Service Framework successfully disconnects the user, the response is a RADIUS-formatted packet with a Disconnect-ACK message.
- If the AAA Service Framework cannot disconnect the user, the request is malformed, or attributes are missing from the request, the response is a RADIUS-formatted packet with a Disconnect-NAK message.

NOTE: The AAA Service Framework processes one dynamic request at a time per subscriber. If the framework receives a second dynamic request while processing a previous request (either a CoA or another Disconnect-Request) for the same subscriber, the framework responds with a Disconnect-NAK message.

Benefits of Radius-Initiated Disconnects

Enables a RADIUS server to dynamically terminate subscriber sessions. This centralized subscriber management feature simplifies handling large numbers of subscribers because operator termination would otherwise require action on the router.

SEE ALSO

Using RADIUS Dynamic Requests for Subscriber Access Management | 3

Configuring RADIUS-Initiated Dynamic Request Support | 4

Usage Thresholds for Subscriber Services

Starting in Junos OS Release 14.1, subscriber management enables you to manage subscriber services by establishing usage thresholds when a service is dynamically activated or when an existing service is modified by a RADIUS CoA action. The service is deactivated when the specified threshold is reached.

Subscriber management supports two types of usage thresholds—traffic volume and time. You use Juniper Networks VSAs to set the usage thresholds. The VSAs are transmitted in RADIUS Access-Accept messages for dynamically activated services, or in RADIUS-initiated CoA-Request messages for existing services. The volume threshold sets the maximum amount of the total input and output traffic that can use the service before the service is deactivated. A time threshold sets the maximum length of time that the service can be active. Table 3 on page 12 shows the VSAs used for volume and time thresholds.

Table 3: Juniper Network VSAs Used for Service Thresholds

Attribute Number	Attribute Name	Description	Value
26-67	Service-Volume	Amount of input and output traffic, in MB, that can use the service; service is deactivated when the volume is exceeded. Tagged VSA, which supports 8 tags (1-8). The router polls the traffic in 10 minute intervals.	• range = 0 through 16777215 MB • 0 = no limit
26-68	Service-Timeout	Number of seconds that the service can be active; service is deactivated when the timeout expires. Tagged VSA, which supports 8 tags (1-8).	• range = 0 through 16777215 seconds • 0 = no timeout

Table 3: Juniper Network VSAs Used for Service Thresholds *(Continued)*

Attribute Number	Attribute Name	Description	Value
26-179	Service-Volume-Gigawords	Amount of input and output traffic, in 4GB units, that can use the service; service is deactivated when the volume is exceeded. Tagged VSA, which supports 8 tags (1-8). The router polls the traffic in 10 minute intervals.	• range = 0 through 16777215 4GB units • 0 = no limit
26-180	Update-Service	New values of service and time quotas for an existing service. Tagged VSA, which supports 8 tags (1-8).	string: *service-name*

SEE ALSO

RADIUS-Initiated Change of Authorization (CoA) Overview | 6

Subscriber Session Logins and Service Activation Failures Overview

IN THIS SECTION

Service and Network Family Activation Process | 15

When a subscriber attempts to log in and is authenticated by RADIUS, the Access-Accept message may include services in the RADIUS Activate-Service VSA (26-65) to be activated for a particular network family. Depending on the configuration and service type, failure to activate a service can result in denial of the subscriber login.

You can use the `service activation` statement at the [`edit access profile` *`profile-name`* `radius options`] hierarchy level to configure the behavior subsequent to an activation failure.

Use the following options to configure this behavior separately for two types of services:

- `dynamic-profile`—This service type is configured in the dynamic profile that is applied by the subscriber access profile.

- `extensible-service`—This service type is configured in an Extensible Subscriber Services Manager (ESSM) operation script. These services often configure new interfaces for business subscribers.

Use the following options to specify whether successful activation of these services is required or optional for subscriber login access:

- `required-at-login`—Activation is required. Failure for any reason causes the Network-Family-Activate-Request for that network family to fail. If no other network family is already active for the subscriber, then the client application logs out the subscriber. This is the default behavior for the `dynamic-profile` service type.

- `optional-at-login`—Activation is optional. Failure due to configuration errors does not prevent activation of the address family; it allows subscriber access. Failure for any other reason causes network family activation to fail. If no other network family is already active for the subscriber, then the client application logs out the subscriber. This is the default behavior for the `extensible-service` service type.

NOTE: Failures associated with the activation of subscriber secure policies (for mirroring traffic to a mediation device) have no effect on access by subscribers subject to the policy.

This configuration does not apply to services activated by means of RADIUS CoA requests, JSRC Push-Profile-Request (PPR) messages, or subscriber secure policies.

For the `dynamic-profile` service type, configuration errors include the following:

- Parsing errors of the dynamic profile and its attributes.
- Missing mandatory user variables.
- References to dynamic profiles that do not exist.
- Semantic check failures in the dynamic profile.

For the `extensible-service` service type, configuration errors include the following:

- Parsing errors of the operation script.
- Commit failures.

To activate a service, authd sends an activation request for the appropriate services to the subscriber management infrastructure (SMI). For example, if the request is for the IPv4 family, then it requests activation of only the IPv4 services. In turn, the SMI sends requests to the server daemons associated with the service, such as cosd or filterd. The results returned by the daemons determine whether the service activation is a success or a failure.

- When all server daemons report success, then SMI reports success to authd and the service is activated.
- If any server daemon reports a configuration error and no daemons report a nonconfiguration error, then SMI reports a configuration error to authd. The service is not activated, but depending on the configuration, the network family activation may succeed.
- If any server daemon reports a nonconfiguration error, then SMI reports failure to authd and the service is not activated.

Service and Network Family Activation Process

When a subscriber logs in, authd has to activate the corresponding address family after the subscriber is authenticated. The client application, such as DHCP or PPP, can request activation of a single network family, IPv4 or IPv6, or it can sequentially request both families to be activated. Successful network family activation is related to the activation of associated services. The following steps describe the process when authd is configured to use RADIUS for authentication:

1. A subscriber attempts to log in.
    a. The client application requests authentication from authd.
    b. authd sends an Access-Request message to the RADIUS server.
    c. The RADIUS server sends an Access-Accept message to authd that includes the RADIUS Activate-Service VSA (26-65).
    d. authd caches the service activation attributes and sends a grant to the client application.
2. The client application sends the first Network-Family-Activate request, for either the IPv4 or IPv6 address family. This request is sometimes referred to as the client-activate request.
3. authd reviews the cached service activation attributes and sends an activation request for the appropriate services to the subscriber management infrastructure (SMI). For example, if the request is for the IPv4 family, then it requests activation of only the IPv4 services. In turn, the SMI sends requests to the server daemons associated with the service, such as cosd or filterd.
4. What authd does next depends on whether the service activation request fails and whether the service is optional or required.
    - When the service activation fails due to a configuration error and the service is optional:
        a. authd deletes the cached service activation attributes for the service.

> **NOTE**: This deletion enables you to re-issue the service request by using a RADIUS change of authorization (CoA) request or a CLI command, without interference from the failed service.

b. authd sends an ACK in response to the family activation request and the family is activated.

c. The subscriber login proceeds.

- When the service activation fails due to a configuration error and the service is required:

a. authd deletes the cached service activation attributes for the service.

> **NOTE**: This deletion enables you to re-issue the service request by using a RADIUS change of authorization (CoA) request or a CLI command, without interference from the failed service.

b. authd sends a NACK in response to the family activation, which causes the client application to terminate the subscriber's login.

- When the service activation fails due to a nonconfiguration error and the service is either optional or required:

a. authd deletes the cached service activation attributes for the service.

> **NOTE**: This deletion enables you to re-issue the service request by using a RADIUS change of authorization (CoA) request or a CLI command, without interference from the failed service.

b. authd sends a NACK in response to the family activation, which causes the client application to terminate the subscriber's login.

- When the service activation succeeds:

a. authd activates the service.

b. authd sends an ACK in response to the family activation request and the family is activated.

c. The subscriber login proceeds.

5. Unless service activation was required and failed, causing the family activation to fail in the first request, the client application may send a second request, but only for the family not requested the first time. If the first request was for IPv4, then the second request can only be for IPv6. If the first request was for IPv6, then the second request can only be for IPv4.

6. authd reviews the cached service activation attributes and requests activation for the services associated with the requested address family.

7. What authd does next depends on whether the service activation request fails and whether the service is optional or required.

   - When the service activation fails due to a configuration error and the service is optional:

     a. authd deletes the cached service activation attributes for the service.

     > **NOTE**: This deletion enables you to re-issue the service request by using a RADIUS change of authorization (CoA) request or a CLI command, without interference from the failed service.

     b. authd sends an ACK in response to the family activation request and the family is activated.

     c. The subscriber login proceeds.

   - When the service activation fails due to a configuration error and the service is required:

     a. authd deletes the cached service activation attributes for the service.

     > **NOTE**: This deletion enables you to re-issue the service request by using a RADIUS change of authorization (CoA) request or a CLI command, without interference from the failed service.

     b. authd sends a NACK in response to the family activation. Because this is the second family activation request, the result of the first family activation determines what happens next:

        - If the first family activation was successful and that subscriber logged in, failure of the second request does not halt the current subscriber login. This event also does not cause authd to log out the previous (first request) subscriber.

        - If the first family activation was unsuccessful, failure of the second request causes the client application to terminate the current subscriber login.

   - When the service activation fails due to a nonconfiguration error and the service is either optional or required:

     a. authd deletes the cached service activation attributes for the service.

> **NOTE**: This deletion enables you to re-issue the service request by using a RADIUS change of authorization (CoA) request or a CLI command, without interference from the failed service.

b. authd sends a NACK in response to the family activation, which causes the client application to terminate the subscriber's login.

- When the service activation succeeds:

  a. authd activates the service.

  b. authd sends an ACK in response to the family activation request and the family is activated.

  c. The subscriber login proceeds.

Configuring How Service Activation Failures Affect Subscriber Login

You can configure how an activation failure of optional services during subscriber login affects the outcome of the login. These optional services are those referenced by the RADIUS Activate-Service VSA (26-65) that appears in the RADIUS Access-Accept message during the subscriber's initial login.

You can configure these two service-activation types to be required or optional.

- `dynamic-profile`—These services are configured in the dynamic profile that is applied by the subscriber access profile to provide subscriber access and services for broadband applications. By default, service activation is required for successful login. A configuration error during service activation prevents the network family from being activated and causes the subscriber login to fail.
- `extensible-service`—These services are applied by operation scripts handled by the Extensible Subscriber Services Manager (ESSM) daemon (essmd) for business subscribers. By default, service activation is optional for successful subscriber login.

> **NOTE**: The `service-activation` statement configuration affects only activation failures due to configuration errors in the dynamic profile or the ESSM operation script. Failures due to nonconfiguration errors always result in denial of access for the subscriber and termination of the login attempt.

To configure the behavior for dynamic profile services, do one of the following:

- Specify that service activation is optional.

```
[edit access profile profile-name radius options service-activation]
user@host# set dynamic-profile optional-at-login
```

- Specify that service activation is required (the default).

```
[edit access profile profile-name radius options service-activation]
user@host# set dynamic-profile required-at-login
```

To configure the behavior for ESSM services, do one of the following:

- Specify that service activation is required.

```
[edit access profile profile-name radius options service-activation]
user@host# set extensible-service required-at-login
```

- Specify that service activation is optional (the default).

```
[edit access profile profile-name radius options service-activation]
user@host# set extensible-service optional-at-login
```

SEE ALSO

Subscriber Session Logins and Service Activation Failures Overview | 13

Error-Cause Codes (RADIUS Attribute 101) for Dynamic Requests

When a RADIUS-initiated CoA or disconnect operation is unsuccessful, the router includes an error-cause attribute (RADIUS attribute 101) in the CoA-NAK or Disconnect-NAK message that it sends back to the RADIUS server. If the detected error does not map to one of the supported error-cause attributes, the router sends the message without an error-cause attribute. Table 4 on page 19 describes the error-cause codes.

Table 4: Error-Cause Codes (RADIUS Attribute 101)

Code	Value	Description
401	Unsupported attribute	The request contains an attribute that is not supported (for example, a third-party attribute).
402	Missing attribute	A critical attribute (for example, the session identification attribute) is missing from a request.

Table 4: Error-Cause Codes (RADIUS Attribute 101) ***(Continued)***

Code	Value	Description
404	Invalid request	Some other aspect of the request is invalid, such as if one or more attributes are not formatted properly.
503	Session context not found	The session context identified in the request does not exist on the router.
504	Session context not removable	The subscriber identified by attributes in the request is owned by a component that is not supported.
506	Resources unavailable	A request could not be honored due to lack of available NAS resources (such as memory).

SEE ALSO

RADIUS-Initiated Change of Authorization (CoA) Overview | 6

RADIUS-Initiated Disconnect Overview | 10

Verifying RADIUS Dynamic-Request Statistics

IN THIS SECTION

- Purpose | 20
- Action | 21

Purpose

Display RADIUS dynamic request statistics and information.

Action

- To display RADIUS dynamic request statistics:

```
user@host>show network-access aaa statistics dynamic-requests
```

SEE ALSO

CLI Explorer

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, bulk CoA requests are supported to improve the processing efficiency of RADIUS-based subscriber services on the BNG.
15.1R5	Starting in Junos OS Release 15.1R5, CoA-Request retry messages are ignored and no CoA-NAK is sent in response to them.
14.1	Starting in Junos OS Release 14.1, subscriber management enables you to manage subscriber services by establishing usage thresholds when a service is dynamically activated or when an existing service is modified by a RADIUS CoA action.

Service Activation and Deactivation Using the CLI Instead of RADIUS

IN THIS SECTION

- CLI-Activated Subscriber Services | 22
- Local and Remote Service Activation and Deactivation Using the CLI | 23

CLI-Activated Subscriber Services

Subscriber management enables you to use the Junos OS CLI to locally activate and deactivate dynamic subscriber services. CLI-based activation and deactivation provides local control for dynamic subscriber services that is similar to subscriber management's change of authorization (CoA) feature. CoA is considered a remote activation method because the commands, or triggers, are received from a remote server, such as a RADIUS or provisioning server. Both the CoA and CLI-based methods enable you to manage services for subscribers who are currently logged in to the network—you can activate a new service for the subscriber or deactivate a current service.

The CLI-based feature activates the specified service—you cannot use it to modify a subscriber's dynamic profile instantiation or to modify user-defined variables in a dynamic profile. You can, however, include variables that are defined for the service in the dynamic profile.

Subscriber management does not support accounting for CLI-activated subscriber services. Accounting for any service is disabled by default. Therefore when you use the CLI to activate a service, it is activated with accounting disabled, and there is no way to explicitly enable accounting for the service. CLI deactivation of a service previously activated (such as by RADIUS) has no effect on accounting for that service.

CLI-based activation and deactivation is useful in service provider networks that do not use provisioning servers or RADIUS servers to activate and deactivate subscriber services. The local control provided by the CLI-based operations enables service providers to add and remove services for existing subscribers without requiring that the subscriber log out and than log in again to complete the change. For example, a service provider might allow subscribers to log in and initially use the default service, which provides basic features. After the default service is established, the provider might then use CLI-activation to upgrade qualified subscribers to an advanced service, in addition to retaining the initial service. Later, the provider can use CLI-deactivation to terminate the subscriber's advanced service session. The subscriber retains the initial service until the service is deactivated.

CLI-based activation or deactivation of a subscriber service fails if any of the following conditions exist:

- A RADIUS CoA operation or a previous CLI-based activation or deactivation is currently in progress for the subscriber. Only one dynamic request can be active for the subscriber.
- A unified in-service software upgrade (unified ISSU) operation is active.
- The specified service could not be activated or deactivated.

A CLI-based activation or deactivation of a subscriber service also fails if a PCRF has successfully activated any services for the subscriber. You must override the PCRF provisioning to be able to activate or deactivate services for such a subscriber. For more information, see *Disabling PCRF Control of a Subscriber Session*.

SEE ALSO

Local and Remote Service Activation and Deactivation Using the CLI

Using the CLI to Modify Traffic-Control Profiles That Are Currently Applied to Subscribers

Default Subscriber Service Overview

Local and Remote Service Activation and Deactivation Using the CLI

Subscriber management enables you to use the Junos OS CLI to locally activate or deactivate dynamic subscriber services for subscribers who are currently logged in to the network. You can activate an initial service for the subscriber, provide an additional service, or deactivate the subscriber's current service. This method is an alternative to using external actions by your RADIUS server.

Starting in Junos OS Release 18.3R1, when the dynamic service profile is configured with the `profile-type remote-device-service` statement, the CLI statements trigger the remote device services manager (RDSM) to provision or deprovision the service on a remote device.

NOTE: A CLI-based activation or deactivation of a subscriber service fails if any of the following conditions exist:

- A RADIUS CoA operation or a previous CLI-based activation or deactivation is active for the subscriber.
- A unified in-service software upgrade (unified ISSU) operation is active.
- The specified service could not be activated or deactivated.

A CLI-based activation or deactivation of a subscriber service also fails if a PCRF has successfully activated any services for the subscriber. You must override the PCRF provisioning to be able to activate or deactivate services for such a subscriber. For more information, see *Disabling PCRF Control of a Subscriber Session*.

However, this caveat does not apply if the service was provisioned on a remote device by the RDSM at the request of PCRF as the external authority supplying the service information. When you issue the command to activate or deactivate such a service, RDSM handles the service action.

To use the CLI to activate a subscriber service:

1. (Optional) Verify the subscriber's ID, and ensure that provisioning is not enabled. To display the session IDs of all current subscribers, use the `show subscribers detail` or `show network-access aaa subscribers` command.

```
user@host> show network-access aaa subscribers session-id 55 detail
Type: dhcp
```

```
Username: user23@example.net
Stripped username: user23
AAA Logical system/Routing instance: default:default
Target Logical system/Routing instance: default:retail-onlinecompany-ca
Access-profile:retailer-onlinecompany-sjc
Session ID: 55
Accounting Session ID: 55
Multi Accounting Session ID: 0
IP Address: 192.168.44.104
Authentication State: AuthStateActive
Accounting State: Acc-Start-Send
Provisioning-type: none
Service name: basic-service
 Service State: SvcActive
 Session ID: 56
 Session uptime: 00:01:45
```

2. Activate the service for the subscriber.

```
user@host> request network-access aaa subscriber add session-id 55 service-profile gold-
service
```

3. (Optional) Verify that the new service is activated for the subscriber. (The initial `basic-service` is also listed because it has not been deactivated.)

```
user@host> show network-access aaa subscribers session-id 55 detail
Type: dhcp
Username: user23@example.net
Stripped username: user23
AAA Logical system/Routing instance: default:default
Target Logical system/Routing instance: default:retail-onlinecompany-ca
Access-profile:retailer-onlinecompany-sjc
Session ID: 55
Accounting Session ID: 55
Multi Accounting Session ID: 0
IP Address: 192.168.44.104
Authentication State: AuthStateActive
Accounting State: Acc-Start-Send
Provisioning-type: none
Service name: basic-service
 Service State: SvcActive
 Session ID: 56
```

```
 Session uptime: 00:02:15
Service name: gold-service
 Service State: SvcActive
 Session ID: 57
 Session uptime: 00:00:30
```

To use the CLI to deactivate a subscriber service:

1. Display the active services for the specified subscriber. The following example shows that the `basic-service` and `gold-service` are active.

```
user@host> show network-access aaa subscribers session-id 55 detail
Type: dhcp
Username: user23@example.net
Stripped username: user23
AAA Logical system/Routing instance: default:default
Target Logical system/Routing instance: default:retail-onlinecompany-ca
Access-profile:retailer-onlinecompany-sjc
Session ID: 55
Accounting Session ID: 55
Multi Accounting Session ID: 0
IP Address: 192.168.44.104
Authentication State: AuthStateActive
Accounting State: Acc-Start-Send
Provisioning-type: none
Service name: basic-service
 Service State: SvcActive
 Session ID: 56
 Session uptime: 00:02:15
Service name: gold-service
 Service State: SvcActive
 Session ID: 57
 Session uptime: 00:00:30
```

2. Deactivate the service for the subscriber. The following example deletes the subscriber's `basic-service` service.

```
user@host> request network-access aaa subscriber delete session-id 55 service-profile basic-
service
```

3. (Optional) Verify that the deleted service is no longer active for the subscriber. (The `gold-service` is still listed because it has not been deactivated.)

```
user@host> show network-access aaa subscribers session-id 55 detail
Type: dhcp
Username: user23@example.net
Stripped username: user23
AAA Logical system/Routing instance: default:default
Target Logical system/Routing instance: default:retail-onlinecompany-ca
Access-profile:retailer-onlinecompany-sjc
Session ID: 55
Accounting Session ID: 55
Multi Accounting Session ID: 0
IP Address: 192.168.44.104
Authentication State: AuthStateActive
Accounting State: Acc-Start-Send
Provisioning-type: none
Service name: gold-service
 Service State: SvcActive
 Session ID: 57
 Session uptime: 00:00:30
```

The following sample commands illustrate CLI activation and deactivation for remote services applied by RDSM to remote devices.

- ```
  user@host> request network-access aaa subscriber add session-id 131 service-profile
  “upstreamBandwidth(100,100,100)”
  Successful completion
  ```

- ```
  user@host> request network-access aaa subscriber delete session-id 131 service-profile
  “upstreamBandwidth(100,100,100)”
  Successful completion
  ```

SEE ALSO

CLI-Activated Subscriber Services

Using the CLI to Modify Traffic-Control Profiles That Are Currently Applied to Subscribers

Default Subscriber Service Overview

Configuring Remote Device Management for Service Provisioning | 715

Remote Device Services Manager (RDSM) Overview | 696

Management of Subscriber Services with Multiple Instances

IN THIS SECTION

- Subscriber Services with Multiple Instances Overview | 27
- Deactivating a Single Instance of a Subscriber Service | 29
- Deactivating All Instances of a Subscriber Service | 33
- Verifying Subscriber Services with Multiple Instances | 36

Subscriber Services with Multiple Instances Overview

IN THIS SECTION

- Subscriber Service Instances and Service Parameters | 27
- CLI Deactivation of Subscriber Services with Multiple Instances | 28
- Subscriber Services with Multiple Instances in RADIUS Accounting Messages | 28

Services are activated for subscribers either at login, or by using Change of Authorization (CoA) RADIUS messages or command-line interface (CLI) requests. A subscriber can have multiple instances of the same named service, provided that each instance of the subscriber service has a different set of parameters. Support for multiple instances of a subscriber service enables you to use service parameters to customize the same service to meet different needs for a particular subscriber.

Subscriber Service Instances and Service Parameters

In a subscriber access network, each subscriber has its own set of services. You can configure a specific *service instance* for a particular subscriber by specifying a *service name*, also referred to as a *service profile*, and unique service parameters for that service instance. *Service parameters* can include a combination of policy lists, filters, rate-limit profiles, *class of service* (CoS) profiles, and interface profiles.

For example, filter-service(up-filter,down-filter) and filter-service(upstream-filter,downstream-filter) are considered two different instances of the same service (filter-service) because their parameters, enclosed in parentheses after the service name, are different.

Each service instance is uniquely identified by the combination of its service name and service parameters. In CoA messages, the router identifies a subscriber service by its complete activation string, which consists of the service name and, if configured, one or more service parameters in the order specified.

CLI Deactivation of Subscriber Services with Multiple Instances

You can use the Junos OS CLI to deactivate subscriber services with multiple instances in either of the following ways:

- Deactivate a single instance of a subscriber service by specifying the name and parameters of the service to be deactivated.

  With this feature, you can deactivate a particular instance of a subscriber service while other instances of that same service remain active. For example, assume that a subscriber identified by a particular session ID has two instances of filter-service activated: filter-service(up-filter,down-filter) and filter-service(upstream-filter,downstream-filter). If you specify "filter-service(up-filter,down-filter)" in the `request network-access aaa subscriber delete session-id` command, the router deactivates only filter-service(up-filter,down-filter); filter-service(upstream-filter,downstream-filter) remains active.

  The ability to use both service names and service parameters to identify the particular service instance to be deactivated is analogous to the subscriber service deactivation feature in use on Juniper Networks E Series Broadband Services Routers that run JunosE Software.

- Deactivate all instances of a subscriber service by specifying only the name of the service to be deactivated, with no service parameters.

  With this feature, you can deactivate all instances of the same subscriber service with a single operational command. Using the same subscriber service example, if you specify "filter-service" in the `request network-access aaa subscriber delete session-id` command, the router deactivates both filter-service(up-filter,down-filter) and filter-service(upstream-filter,downstream-filter).

Subscriber Services with Multiple Instances in RADIUS Accounting Messages

RADIUS Acct-Start, Interim-Acct, and Acct-Stop accounting messages include the subscriber service name and, if configured, service parameters. If RADIUS logging is enabled, the router logs all subscriber service attributes, including service names and parameters, in messages sent to and received from the RADIUS authentication server.

For example, assume that the router receives the following RADIUS Access-Accept message from the RADIUS server:

```
Jul 13 12:37:02 radius-access-accept: Activate-Service (Juniper-ERX-VSA) received: Tag (1)
filter-service(up-filter,down-filter)
```

Table 5 on page 29 shows sample logged RADIUS Acct-Start, Interim-Acct, and Acct-Stop messages that the router sends to the RADIUS server in response to the Access-Accept message. In each of these accounting messages, the Activate-Service-Session-Name is the full activation string that includes both the service name (filter-service) and service parameters (up-filter,down-filter) to identify the service instance.

Table 5: Subscriber Services and Service Parameters in RADIUS Accounting Messages

RADIUS Accounting Message Type	RADIUS Accounting Message Text
Acct-Start	`Jul 13 12:37:02 radius-acct-start: Activate-Service-Session-Name (Juniper-ERX-VSA) added: filter-service(up-filter,down-filter)`
Interim-Acct	`Jul 13 12:47:00 radius-acct-interim: Activate-Service-Session-Name (Juniper-ERX-VSA) added: filter-service(up-filter,down-filter)`
Acct-Stop	`Jul 13 12:53:59 radius-acct-stop: Activate-Service-Session-Name (Juniper-ERX-VSA) added: filter-service(up-filter,down-filter)`

SEE ALSO

Deactivating a Single Instance of a Subscriber Service | 29

Deactivating All Instances of a Subscriber Service | 33

Verifying Subscriber Services with Multiple Instances | 36

Deactivating a Single Instance of a Subscriber Service

For subscriber services that have multiple instances, you can use the Junos OS CLI to deactivate a service in either of the following ways:

- Deactivate a single instance of a service by specifying the name and parameters of the service to be deactivated.
- Deactivate all instances of a service by specifying only the name of the service to be deactivated.

This topic describes how to deactivate a single instance of a subscriber service.

To use the Junos OS CLI to deactivate a single instance of a subscriber service with multiple instances:

1. Display the active services for the subscriber identified by the specified session ID.

```
user@host> show network-access aaa subscribers session-id subscriber-session-id detail
```

For example, the following command displays the active services for the DHCP subscriber identified by session ID 6. In this example, two instances of economy-service are active: `economy-service(up-filter,down-filter)` and `economy-service(upstrm-filter,dwnstrm-filter)`. A single instance of premium-service named `premium-service(up-filter,down-filter)` is also active.

```
user@host> show network-access aaa subscribers session-id 6 detail
Type: dhcp
Stripped username: fran2
AAA Logical system/Routing instance: default:default
Target Logical system/Routing instance: default:default
Access-profile: attr_test_profile1
Session ID: 6
Accounting Session ID: 6
Multi Accounting Session ID: 0
IP Address: 198.51.100.10
Authentication State: AuthStateActive
Accounting State: Acc-Interim-Sent
Provisioning Type: None
Service name: economy-service(up-filter,down-filter)
  Service State: SvcActive
  Service Family: inet
  Service Activation Source: Radius
  Session ID: 7
  Session uptime: 00:04:36
  Accounting status: on/volume+time
  Service accounting session ID: 6:7-1354811427
  Service accounting state: Acc-Start-Sent
  Accounting interim interval: 600
Service name: economy-service(upstrm-filter,dwnstrm-filter)
  Service State: SvcActive
```

```
  Service Family: inet
  Service Activation Source: Radius
  Session ID: 8
  Session uptime: 00:04:36
  Accounting status: on/volume+time
  Service accounting session ID: 6:8-1354811427
  Service accounting state: Acc-Start-Sent
  Accounting interim interval: 600
Service name: premium-service(up-filter,down-filter)
  Service State: SvcActive
  Service Family: inet
  Service Activation Source: Radius
  Session ID: 9
  Session uptime: 00:04:36
  Accounting status: on/volume+time
  Service accounting session ID: 6:9-1354811427
  Service accounting state: Acc-Start-Sent
  Accounting interim interval: 600
```

2. Deactivate the specified instance of a subscriber service by specifying its service name and parameters.

```
user@host> request network-access aaa subscriber delete session-id  subscriber-session-id
service-profile “profile-name(parameters)”
```

For example, the following command deactivates only the instance of economy-service named economy-service(up-filter,down-filter).

```
user@host> request network-access aaa subscriber delete session-id  6 service-profile
“economy-service(up-filter,down-filter)”
```

3. (Optional) Verify that the deactivated service instance is no longer active for the subscriber.

```
user@host> show network-access aaa subscribers session-id subscriber-session-id detail
```

For example, the following command displays the services still active for the DHCP subscriber identified by session ID 6. In this example, `economy-service(up-filter,down-filter)` is no longer listed

because it was deactivated, but `economy-service(upstrm-filter,dwnstrm-filter)` and `premium-service(up-filter,down-filter)` are still active.

```
user@host> show network-access aaa subscribers session-id 6 detail
Type: dhcp
Stripped username: fran2
AAA Logical system/Routing instance: default:default
Target Logical system/Routing instance: default:default
Access-profile: attr_test_profile1
Session ID: 6
Accounting Session ID: 6
Multi Accounting Session ID: 0
IP Address: 198.51.100.13.10
Authentication State: AuthStateActive
Accounting State: Acc-Interim-Sent
Provisioning Type: None
Service name: economy-service(upstrm-filter,dwnstrm-filter)
  Service State: SvcActive
  Service Family: inet
  Service Activation Source: Radius
  Session ID: 8
  Session uptime: 00:04:36
  Accounting status: on/volume+time
  Service accounting session ID: 6:8-1354811427
  Service accounting state: Acc-Start-Sent
  Accounting interim interval: 600
Service name: premium-service(up-filter,down-filter)
  Service State: SvcActive
  Service Family: inet
  Service Activation Source: Radius
  Session ID: 9
  Session uptime: 00:04:36
  Accounting status: on/volume+time
  Service accounting session ID: 6:9-1354811427
  Service accounting state: Acc-Start-Sent
  Accounting interim interval: 600
```

SEE ALSO

Deactivating All Instances of a Subscriber Service | 33

Verifying Subscriber Services with Multiple Instances | 36

Subscriber Services with Multiple Instances Overview | 27

Deactivating All Instances of a Subscriber Service

For subscriber services that have multiple instances, you can use the Junos OS CLI to deactivate a service in either of the following ways:

- Deactivate a single instance of a service by specifying the name and parameters of the service to be deactivated.
- Deactivate all instances of a service by specifying only the name of the service to be deactivated.

This topic describes how to deactivate all instances of a subscriber service.

To use the Junos OS CLI to deactivate all instances of a subscriber service with multiple instances:

1. Display the active services for the subscriber identified by the specified session ID.

```
user@host> show network-access aaa subscribers session-id subscriber-session-id detail
```

For example, the following command displays the active services for the DHCP subscriber identified by session ID 6. In this example, two instances of economy-service are active: `economy-service(up-filter,down-filter)` and `economy-service(upstrm-filter,dwnstrm-filter)`. A single instance of premium-service named `premium-service(up-filter,down-filter)` is also active.

```
user@host> show network-access aaa subscribers session-id 6 detail
Type: dhcp
Stripped username: fran2
AAA Logical system/Routing instance: default:default
Target Logical system/Routing instance: default:default
Access-profile: attr_test_profile1
Session ID: 6
Accounting Session ID: 6
Multi Accounting Session ID: 0
IP Address: 198.51.100.10
Authentication State: AuthStateActive
Accounting State: Acc-Interim-Sent
Provisioning Type: None
Service name: economy-service(up-filter,down-filter)
  Service State: SvcActive
  Service Family: inet
  Service Activation Source: Radius
```

```
  Session ID: 7
  Session uptime: 00:04:36
  Accounting status: on/volume+time
  Service accounting session ID: 6:7-1354811427
  Service accounting state: Acc-Start-Sent
  Accounting interim interval: 600
Service name: economy-service(upstrm-filter,dwnstrm-filter)
  Service State: SvcActive
  Service Family: inet
  Service Activation Source: Radius
  Session ID: 8
  Session uptime: 00:04:36
  Accounting status: on/volume+time
  Service accounting session ID: 6:8-1354811427
  Service accounting state: Acc-Start-Sent
  Accounting interim interval: 600
Service name: premium-service(up-filter,down-filter)
  Service State: SvcActive
  Service Family: inet
  Service Activation Source: Radius
  Session ID: 9
  Session uptime: 00:04:36
  Accounting status: on/volume+time
  Service accounting session ID: 6:9-1354811427
  Service accounting state: Acc-Start-Sent
  Accounting interim interval: 600
```

2. Deactivate all instances of the specified service by specifying the service name without parameters.

```
user@host> request network-access aaa subscriber delete session-id  subscriber-session-id
service-profile “profile-name”
```

For example, the following command deactivates both instances of economy-service.

```
user@host> request network-access aaa subscriber delete session-id  6 service-profile
“economy-service”
```

3. (Optional) Verify that all instances of the deactivated service are no longer active for the subscriber.

```
user@host> show network-access aaa subscribers session-id subscriber-session-id detail
```

In the following example, only `premium-service(up-filter,down-filter)` is still active. Neither `economy-service(up-filter,down-filter)` nor `economy-service(upstrm-filter,dwnstrm-filter)` is listed because all instances of economy-service were deactivated.

```
user@host> show network-access aaa subscribers session-id 6 detail
Type: dhcp
Stripped username: fran2
AAA Logical system/Routing instance: default:default
Target Logical system/Routing instance: default:default
Access-profile: attr_test_profile1
Session ID: 6
Accounting Session ID: 6
Multi Accounting Session ID: 0
IP Address: 198.51.100.10
Authentication State: AuthStateActive
Accounting State: Acc-Interim-Sent
Provisioning Type: None
Service name: premium-service(up-filter,down-filter)
  Service State: SvcActive
  Service Family: inet
  Service Activation Source: Radius
  Session ID: 9
  Session uptime: 00:04:36
  Accounting status: on/volume+time
  Service accounting session ID: 6:9-1354811427
  Service accounting state: Acc-Start-Sent
  Accounting interim interval: 600
```

SEE ALSO

- Subscriber Services with Multiple Instances Overview | 27
- Deactivating a Single Instance of a Subscriber Service | 29
- Verifying Subscriber Services with Multiple Instances | 36

Verifying Subscriber Services with Multiple Instances

IN THIS SECTION

- Purpose | 36
- Action | 36
- Meaning | 37

Purpose

Display information about the active services for a subscriber identified by the specified session ID.

Action

The following example displays information about the active services for the DHCP subscriber identified by session ID 6.

```
user@host> show network-access aaa subscribers session-id 6 detail
Type: dhcp
Stripped username: fms2
AAA Logical system/Routing instance: default:default
Target Logical system/Routing instance: default:default
Access-profile: attr_test_profile1
Session ID: 6
Accounting Session ID: 6
Multi Accounting Session ID: 0
IP Address: 198.51.100.10
Authentication State: AuthStateActive
Accounting State: Acc-Interim-Sent
Provisioning Type: None
Service name: economy-service(up-filter,down-filter)
  Service State: SvcActive
  Service Family: inet
  Service Activation Source: Radius
  Session ID: 7
  Session uptime: 00:04:36
  Accounting status: on/volume+time
  Service accounting session ID: 6:7-1354811427
```

```
  Service accounting state: Acc-Start-Sent
  Accounting interim interval: 600
Service name: economy-service(upstrm-filter,dwnstrm-filter)
  Service State: SvcActive
  Service Family: inet
  Service Activation Source: Radius
  Session ID: 8
  Session uptime: 00:04:36
  Accounting status: on/volume+time
  Service accounting session ID: 6:8-1354811427
  Service accounting state: Acc-Start-Sent
  Accounting interim interval: 600
Service name: premium-service
  Service State: SvcActive
  Service Family: inet
  Service Activation Source: Radius
  Session ID: 9
  Session uptime: 00:04:36
  Accounting status: on/volume+time
  Service accounting session ID: 6:9-1354811427
  Service accounting state: Acc-Start-Sent
  Accounting interim interval: 600
```

Meaning

If parameters are configured when a subscriber service with multiple instances is activated, the `Service name` field in the `show network-access aaa subscribers session-id` command displays both the service name and, in parentheses following the service name, the service parameters. If parameters are not configured for a particular service, the `show network-access aaa subscribers session-id` command displays only the service name. The value `SvcActive` in the `Service State` field indicates that the service is active.

In this example, two instances of economy-service are active: `economy-service(up-filter,down-filter)` and `economy-service(upstrm-filter,dwnstrm-filter)`. For `premium-service`, which is also active, the command output displays only the service name, indicating that no parameters were configured for this service.

SEE ALSO

- Subscriber Services with Multiple Instances Overview | 27
- Deactivating a Single Instance of a Subscriber Service | 29
- Deactivating All Instances of a Subscriber Service | 33

Subscriber Management Redundancy for PFE During Graceful OIR

SUMMARY

Subscriber Management Support for PFE Graceful Online Insertion and Removal on Line card MICs (MX304-LMIC).

Use subscriber management redundancy on the Packet Forwarding Engine (PFE) for seamless online insertion and removal (OIR) on MX304 devices. Subscribers and flows are retained when an alternate PFE provides redundancy. This feature support includes the following functionalities:

- When a PFE is made offline, all subscribers hosted on it are terminated, and their flows are deleted, except if there is redundancy provided by an alternate PFE. However, DHCP subscribers remain UP even if PFE is made offline. If the line card MIC (MX304-LMIC) comes back online, the pre-existing DHCP subscriber functionalities resume.
- When a PFE goes offline, its subsidiary links are removed. If an aggregate interface has active members on another PFE, the subscriber flows remain intact. Statistics for terminated subscribers are reported as the PFE goes offline.
- No new subscriber states are activated on the offline PFE. Other PFEs manage subscriber logins and logouts. The newly online PFE integrates seamlessly with other online PFEs to maintain subscriber management services.
- There is a temporary delay in forwarding configuration for new subscribers during PFE transitions, depending on the number of active subscribers.
- CPU utilization spikes during the offline process due to subscriber state transitions and during the online process if there are many subscribers.
- All interfaces corresponding to LMIC being offline are transitioned DOWN. Subscriber management is active only on online PFEs.
- The accounting statistics for subscribers are cached when PFE goes offline. The accounting statistics for the subscribers provide the accurate values across offline-online activity of the associated PFE. When PFE goes offline, the interface statistics are cleared for the subscriber.
- The `set chassis fpc <> pfe <> power <on | off>` command is not supported on MX304 devices.
- The command `request chassis mic fpc-slot <> mic-slot <> <offline | online>` is supported is supported on MX304 for graceful OIR.

- [See request chassis mic .]

The following tables describe the OIR support and impact related to various features and functionalities such as subscriber types, firewall behavior, BFS, Layer-2 and Layer-2 functionalities and telemetry.

Table 6: Subscriber Types during PFE OIR

Functionality	PFE Handling
Subscribers on Gigabit Ethernet/10 Gigabit Ethernet physical interfaces:	• When PFE goes offline, all PPPOE subscribers on those interfaces are deleted. • When the PFE is online, new subscribers can log in.
Aggregate Ethernet IFDs:	• When PFE goes offline, associated members of the Aggregate Ethernet interface are r Subscribers on active members hosted by other PFEs remain active. • When the PFE comes back online, the subscriber states are restored.
L2TP Subscribers:	• Subscribers over Service Interfaces (SI) on device such as Logistics Network Services (Logistics Transport Services (LTS) are deleted when the PFE goes offline. • If the PFE is part of the L2TP forwarding path, the sessions might deactivate due to k issues.
Subscribers over PS (Pseudowire Subscriber Interfaces):	On Gigabit Ethernet/10 Gigabit Ethernet interfaces the subscribers over PS are deleted w PFE goes offline.

Table 7: Layer 2 , Layer 3, and BFD during PFE OIR

Functionality	PFE Handling
Bidirectional Forwarding Detection (BFD):	• BFD sessions configured on the anchor PFE are disabled if the PFE goes offline ar disabled until the PFE comes back online. • Non-anchor PFEs clean up session states when going offline and reprogram the se when brought back online.

Table 7: Layer 2 , Layer 3, and BFD during PFE OIR *(Continued)*

Functionality	PFE Handling
Layer 2:	• Bridge domains and Layer 2 token management are unaffected, as their states are per-PFE. • **CFM (Connectivity Fault Management)** sessions are reassigned to another PFE if t original PFE goes offline, that can result in a temporary interruption.
Layer 3:	For routing protocols and related features (like OSPF or BFD), the PFE offline/online a trigger necessary reprogramming.

Broadband Network Gateway (BNG) Subscriber Redundancy on Aggregated Ethernet (AE) Interfaces with Disabled PFEs (MX304, MX960, and MX10004)

SUMMARY

The BNG subscriber redundancy feature on Aggregated Ethernet (AE) interfaces enhances network reliability by ensuring uninterrupted subscriber sessions during Packet Forwarding Engine (PFE) disable.

IN THIS SECTION

- Benefits of BNG Subscriber Resilience on Aggregated Ethernet Interfaces with Disabled PFEs | **41**

Broadband Network Gateway subscriber resiliency on aggregated Ethernet interfaces with disabled PFEs (MX304, MX960, MX10004)—You can enhance network reliability for Broadband Network Gateway (BNG) subscribers by maintaining operational status over aggregated Ethernet (AE) interfaces and ensuring uninterrupted subscriber sessions even when a Packet Forwarding Engine (PFE) is disabled. This feature ensures uninterrupted subscriber sessions using Dynamic Host Configuration Protocol (DHCP), Point-to-Point Protocol over Ethernet (PPPoE), or Layer 2 Tunneling Protocol (L2TP) as long as one member link remains active. It supports both Active/Active and Active/Standby modes and is compatible with dynamic and static VLAN configurations.

The BNG subscriber redundancy feature on AE interfaces enhances network reliability by ensuring uninterrupted subscriber sessions even when a PFE is disabled. This feature maintains active sessions using DHCP, Point-to-Point Protocol over Ethernet (PPPoE), or Layer 2 Tunneling Protocol (L2TP) as long as at least one member link in the AE interface remains active. It supports both Active/Active and

Active/Standby modes and is compatible with dynamic and static VLAN configurations. PPPoE sessions and inline Bidirectional Forwarding Detection (BFD) for DHCP also ensure session integrity.

Benefits of BNG Subscriber Resilience on Aggregated Ethernet Interfaces with Disabled PFEs

- **Enhanced Network Reliability:** Ensures that BNG subscriber sessions remain active and traffic is forwarded uninterrupted, even if a PFE encounters a hardware failure.
- **Service Continuity:** Maintains subscriber connectivity by keeping sessions operational as long as at least one member link in the AE interface is on an active PFE.
- **Minimized Service Disruptions:** Allows for efficient traffic management by preventing complete service outages, thus enhancing user experience and reducing downtime.
- **Compatibility with Various Configurations:** Supports both Active/Active and Active/Standby modes for AE interfaces, and works with dynamic VLAN, static VLAN, and static demultiplexing VLAN configurations, offering flexibility in network design.

PART 2

Configuring Dynamic Class of Service

- CoS for Subscriber Access and Interfaces Overview | **43**
- Configuring Scheduling and Shaping for Subscriber Access | **52**
- Allocating Dedicated Queues for Each Logical Interface Using Per-Unit Scheduling | **74**
- Configuring Hierarchical CoS Scheduling on MPLS Ethernet Pseudowire Subscriber Interfaces | **89**
- Configuring Dedicated Queue Scaling with Hierarchical CoS or Per-Unit Scheduling | **107**
- Shaping Downstream Traffic Based on Frames or Cells | **115**
- Applying CoS to Households or Individual Subscribers Using ACI-Based Dynamic VLANs | **134**
- Applying CoS to Households or Individual Subscribers Using Access Line Identifier Dynamic VLANs | **144**
- Managing Excess Bandwidth Distribution and Traffic Bursts | **154**
- Applying CoS Using Parameters Received from RADIUS | **161**
- Modifying a Subscriber' s Shaping Characteristics After a Subscriber is Instantiated | **189**
- Applying CoS to Groups of Subscriber Interfaces | **196**
- Applying CoS to Subscriber Interfaces | **228**

CHAPTER 2

CoS for Subscriber Access and Interfaces Overview

IN THIS CHAPTER

- CoS for Subscriber Access Overview | 43
- Guidelines for Configuring Dynamic CoS for Subscriber Access | 44
- CoS for Aggregated Ethernet Subscriber Interfaces Overview | 49
- CoS for PPPoE Subscriber Interfaces Overview | 50

CoS for Subscriber Access Overview

This topic describes class-of-service (CoS) functionality for dynamic subscriber access.

Junos CoS enables you to divide traffic into classes and offer various levels of throughput and packet loss when congestion occurs. This functionality allows packet loss to happen according to rules that you configure. The Junos CoS features provide a set of mechanisms that you can use to provide differentiated services when best-effort traffic delivery is insufficient.

In a subscriber access environment, service providers want to provide video, voice, and data services over the same network for subscribers. Subscriber traffic is delivered from the access network, through a router, through a switched Ethernet network, to an Ethernet digital subscriber line access multiplexer (DSLAM). The DSLAM forwards the subscriber's traffic to the residential gateway over a digital subscriber line (DSL). An MX Series router that is installed in a subscriber access network as an edge router can perform subscriber management functions that include subscriber identification and per-subscriber CoS.

In a subscriber access network, a subscriber is an authenticated user—a user that has logged in to the access network at a subscriber interface and then been verified by the configured authentication server and subsequently granted initial CoS services. Subscribers can be identified statically or dynamically. In this network, subscribers are mapped to VLANs, demux, or PPPoE interfaces.

You can configure the router to provide *hierarchical scheduling* or *per-unit scheduling* for subscribers:

- Hierarchical CoS enables you to apply traffic scheduling and queuing parameters (which can include a delay-buffer bandwidth) and packet transmission scheduling parameters (which can include buffer management parameters) to an individual subscriber interface rather than to all interfaces configured

on the port. Hierarchical CoS enables you to dynamically modify queues when subscribers require services.

- Per-unit scheduling enables one set of output queues for each *logical interface* configured under the physical interface. In per-unit scheduling configurations, each Layer 3 scheduler node is allocated a dedicated set of queues.

Because the interface sets corresponding to VLANs using agent-circuit-identifier information are created dynamically, you can apply CoS attributes, such as shaping, at the household level. You must set and define the CoS policy for the agent-circuit-identifier virtual VLAN interface set using the dynamic profile for the agent-circuit-identifier interface set (not the subscriber profile). CoS on dynamic VLANs includes support for level 4, level 3, or level 2 scheduler nodes for a dynamic interface set. You can also configure a traffic-control profile and a remaining traffic-control profile for a dynamic interface set. CoS on dynamic VLANs enables you to configure a dynamic scheduler map for a traffic-control profile that is used by a dynamic interface set. In this case, the dynamic scheduler map must use the unique ID format.

RELATED DOCUMENTATION

Understanding Hierarchical CoS for Subscriber Interfaces

Guidelines for Configuring Dynamic CoS for Subscriber Access | 44

Configuring Static Hierarchical Scheduling in a Dynamic Profile

Configuring Per-Unit Scheduling in a Dynamic Profile | 75

Guidelines for Configuring Dynamic CoS for Subscriber Access

IN THIS SECTION

- Configuration Guidelines for Hierarchical CoS and Per-Unit Scheduling | 45
- Configuration Guidelines for Dynamic Scheduling and Queuing | 45
- Configuration Guidelines for Dynamic Classifiers and Rewrite Rules | 46

This topic describes the guidelines for configuring dynamic CoS in a subscriber access environment.

Configuration Guidelines for Hierarchical CoS and Per-Unit Scheduling

You can configure dynamic CoS with one of the following scheduling configurations:

- For hierarchical scheduling configurations, you must enable hierarchical scheduling in the static CLI for the interface referenced in the dynamic profile. If not, the dynamic profile fails.
- For per-unit scheduling configurations, you must enable per-unit scheduling in the static CLI for the interface referenced in the dynamic profile. If not, the dynamic profile fails and schedulers are not attached to the interface.

Junos software supports either per-unit scheduling or hierarchical scheduling on an interface. You cannot run both types of scheduling at the same time. If CoS is active on an interface, and you change the type of scheduling configured on the interface, all traffic is dropped upon egress from the interface.

Configuration Guidelines for Dynamic Scheduling and Queuing

When configuring scheduling and queuing for subscriber access, consider the following guidelines:

- To improve CoS performance in IPv4, IPv6, and dual-stack networks that use a DHCP access model, we recommend that you use the `aggregate-clients replace` statement rather than the `aggregate-clients merge` statement.
- You configure the traffic scheduling and shaping parameters in a traffic-control profile within the dynamic profile. You can configure the scheduler map and schedulers in a dynamic profile or in the `[edit class-of-service]` hierarchy. You must statically configure the remaining CoS parameters, such as hierarchical scheduling, classifiers, drop profiles, and forwarding classes, in the `[edit class-of-service]` hierarchy.
- You can configure only one traffic-control-profile under a dynamic profile.
- You must define the output-traffic-control-profile that binds the traffic-control profile to the interface within the same dynamic profile as the interface.
- We recommend that you provide different names for the schedulers defined in dynamic profiles that are used for access and services. For example, if there are two dynamic profiles, voice-profile and video-profile, provide unique names for the schedulers defined under those profiles.
- You must use a service dynamic profile with a different profile name for each RADIUS CoA request over the same *logical interface*.
- When you configure scheduler and scheduler map sharing in client profiles, schedulers and scheduler maps must use the unique ID format. If the client profile uses the unique ID format and you want to have either scheduler or scheduler map sharing for service activation, you must configure the service profile in unique ID format.

Configuration Guidelines for Dynamic Classifiers and Rewrite Rules

When you configure classifiers and *rewrite rules* for subscriber access, consider the following guidelines:

- To apply classifiers and rewrite rules to a subscriber interface in a dynamic profile, you must configure the rewrite rule and classifier definitions in the static `[edit class-of-service]` hierarchy and reference them in the dynamic profile.
  - If a static classifier or a rewrite rule definition that is referenced by a dynamic subscriber interface does not exist, the configuration is invalid and the subscriber cannot log in.
  - If a network administrator changes the static classifiers and rewrite rules definitions that are referenced in a dynamic profile with an active subscriber interface logged in, the changes are applied to the active subscriber interface immediately.
  - If a network administrator deletes a classifier or a rewrite rule definition that is referenced by an active dynamic subscriber interface, the system removes the classifier or rewrite rule binding from the interface. The classifier is replaced by the default classifier. If the network administrator adds the removed classifier or rewrite rule to the configuration while the dynamic interface is active, the addition does not take effect until the subscriber logs out and then logs in again.
- IP demux interfaces can only instantiate Layer 3 rules (both rewrite rules and classifiers).
  - An IP demux subscriber interface can implicitly inherit a classifier from the underlying interface. If an IP demux interface is created without a classifier and a Layer 2 classifier is attached to the underlying interface, the IP demux interface also inherits the Layer 2 classifier. The `show class-of-service interface` *`interface-name`* command does not display this attachment.

    Table 8 on page 46 lists the classification rule configuration for an IP demux subscriber interface with a VLAN underlying interface.

    Table 8: IP Demux Classification Rules

VLAN Underlying Interface Classifier Configuration	IP Demux Interface Classifier Configuration	Resulting Classifier Configuration
Layer 2	—	VLAN Layer 2
Layer 2	Layer 3	Demux Layer 3
Layer 3	—	Default

Table 8: IP Demux Classification Rules *(Continued)*

VLAN Underlying Interface Classifier Configuration	IP Demux Interface Classifier Configuration	Resulting Classifier Configuration
Layer 3	Layer 3	Demux Layer 3

- An IP demux subscriber interface explicitly inherits Layer 2 rewrite rules from the underlying interface if a Layer 2 rewrite rule is present. The `show class-of-service interface interface-name` command displays the attachment.

  Table 9 on page 47 lists the rewrite rule configuration for an IP demux subscriber interface with a VLAN underlying interface.

  Table 9: IP Demux Rewrite Rules

VLAN Underlying Interface Rewrite Rule Configuration	IP Demux Interface Rewrite Rule Configuration	Resulting Rewrite Rule Configuration
Layer 2	—	VLAN Layer 2
Layer 2	Layer 3	VLAN Layer 2 and demux Layer 3
Layer 3	—	Default
Layer 3	Layer 3	Demux Layer 3

- An L2TP subscriber interface can implicitly inherit a classifier from the underlying interface.

  Table 10 on page 47 lists the classification rule configuration for an L2TP LAC subscriber interface with a VLAN underlying interface.

  Table 10: L2TP Classification Rules

VLAN Underlying Interface Classifier Configuration	L2TP LAC Classifier Configuration	Resulting Classifier Configuration
Layer 2 or Fixed	Layer 2 or Fixed	VLAN Layer 2 or Fixed

Table 10: L2TP Classification Rules *(Continued)*

VLAN Underlying Interface Classifier Configuration	L2TP LAC Classifier Configuration	Resulting Classifier Configuration
Layer 2 or Fixed	Layer 3	Demux/PPPoE Layer 3
Layer 3	Layer 2 or Fixed	VLAN Layer 2 or Fixed
Layer 3	Layer 3	Demux/PPPoE Layer 3

- An L2TP LAC subscriber interface explicitly inherits Layer 2 rewrite rules from the underlying interface if a Layer 2 rewrite rule is present. Table 11 on page 48 lists the rewrite rule configuration for an L2TP LAC subscriber interface with a VLAN underlying interface.

Table 11: L2TP LAC Rewrite Rules

VLAN Underlying Interface Rewrite Rule Configuration	L2TP Interface Rewrite Rule Configuration	Resulting Rewrite Rule Configuration
Layer 2	Layer 2	VLAN Layer 2
Layer 2	Layer 3	VLAN Layer 2 and demux/PPPoE Layer 3
Layer 3	Layer 2	VLAN Layer 2 and demux/PPPoE Layer 3
Layer 3	Layer 3	Demux/PPPoE Layer 3

RELATED DOCUMENTATION

CoS for Subscriber Access Overview | **43**

Understanding Hierarchical CoS for Subscriber Interfaces

Configuring Static Hierarchical Scheduling in a Dynamic Profile

Configuring Per-Unit Scheduling in a Dynamic Profile | **75**

Configuring Static CoS for an L2TP LNS Inline Service

CoS for Aggregated Ethernet Subscriber Interfaces Overview

You can apply static or dynamic hierarchical CoS to a scheduler node at the aggregated Ethernet *logical interface*, its underlying physical interface, or an interface set.

When you configure CoS for aggregated Ethernet interfaces, consider the following guidelines:

- Configure the aggregated Ethernet logical interface over two physical interfaces capable of performing hierarchical scheduling.
- For VLAN subscriber interfaces over aggregated Ethernet, you must enable link protection on the aggregated Ethernet interface for hierarchical CoS to operate.
- Link protection is not required for IP or demux subscriber interfaces over aggregated Ethernet. We recommend that you enable targeted distribution on the demux interface to provide accurate hierarchical scheduling for these links.
- Keep the following guidelines in mind when configuring interface sets of aggregated Ethernet interfaces:
  - Sets of aggregated Ethernet interfaces are supported on MPC/MIC interfaces on MX Series routers only.
  - The supported logical interfaces for aggregated Ethernet in an interface set include VLAN demux interfaces, IP demux interfaces, and PPPoE logical interfaces over VLAN demux interfaces.
  - The link membership list and scheduler mode of the interface set are inherited from the underlying aggregated Ethernet interface over which the interface set is configured.
  - When an aggregated Ethernet interface operates in link protection mode, or if the scheduler mode is configured to replicate member links, the scheduling parameters of the interface set are copied to each of the member links.
  - If the scheduler mode of the aggregated Ethernet interface is set to scale member links, the scheduling parameters are scaled based on the number of active member links and applied to each of the aggregated interface member links.

BEST PRACTICE: While subscribers are active on aggregated Ethernet physical interfaces with targeted distribution, we recommend that you do not change any attribute of the physical interfaces, such as MTU. Instead, perform the following steps:

1. Log out all the subscribers.
2. Disable the interface.

3. Make the desired attribute changes.

4. Reenable the interface.

If you do not follow these steps, the attribute change brings down the physical interface and all subscribers using that interface.

To avoid service interruptions, we recommend that you make the changes during a maintenance window.

RELATED DOCUMENTATION

- Understanding Hierarchical CoS for Subscriber Interfaces
- Configuring Hierarchical CoS for a Subscriber Interface of Aggregated Ethernet Links
- Configuring an Interface Set of Subscribers in a Dynamic Profile | **199**
- *Static or Dynamic Demux Subscriber Interfaces over Aggregated Ethernet Overview*
- *Static and Dynamic VLAN Subscriber Interfaces over Aggregated Ethernet Overview*
- Distribution of Demux Subscribers in an Aggregated Ethernet Interface

CoS for PPPoE Subscriber Interfaces Overview

For all supported hardware platforms, you can attach an output traffic-control profile that contains basic shaping and scheduling properties directly to a static or dynamic PPPoE interface. In this type of scenario, you can use each PPPoE interface to represent a household and shape all of the household traffic to an aggregate rate. Each forwarding class is mapped to a queue, and represents one type of services provided to a household customer.

For MPCs that support hierarchical scheduling, you can shape subscriber or access node traffic at different levels of the PPPoE interface hierarchy by attaching traffic-control profiles to interface sets that contain PPPoE members.

MPCs support subscriber interfaces with PPPoE encapsulation over aggregated Ethernet interfaces. These PPPoE subscriber interfaces are configured over VLAN demux interfaces, which are also configured over Aggregated Ethernet interfaces.

You can configure 802.3ad link aggregation group (LAG) stateful port and dense port concentrator (DPC) redundancy. This provides targeted distribution of non-replicated (stacked) PPPoE or IP demux links over VLAN demux links, which in turn are over an aggregated Ethernet (AE) *logical interface*. Service

providers with PPPoE or IP demux interfaces for CoS configurations can provide DPC and port redundancy to subscribers.

NOTE: For static PPPoE underlying logical interfaces, use PPPoE interface sets.

RELATED DOCUMENTATION

- Understanding Hierarchical CoS for Subscriber Interfaces
- Configuring Static Hierarchical Scheduling in a Dynamic Profile
- Configuring Hierarchical CoS on a Static PPPoE Subscriber Interface
- CoS on Enhanced IQ2 PICs Overview

CHAPTER 3

Configuring Scheduling and Shaping for Subscriber Access

IN THIS CHAPTER

- Configuring Traffic Scheduling and Shaping for Subscriber Access | 52
- Configuring Schedulers in a Dynamic Profile for Subscriber Access | 58
- Configuring Scheduler and Scheduler Map Sharing | 66
- Example: Providing Unique Rate Configurations for Schedulers in a Dynamic Profile | 68
- Example: Configuring Aggregate Scheduling of Queues for Residential Subscribers on Static IP Demux Interfaces | 70
- Verifying the Scheduling and Shaping Configuration for Subscriber Access | 72

Configuring Traffic Scheduling and Shaping for Subscriber Access

IN THIS SECTION

- Configuring Static Traffic Shaping and Scheduling Parameters in a Dynamic Profile | 53
- Configuring Dynamic Traffic Shaping and Scheduling Parameters in a Dynamic Profile | 54
- Using the CLI to Modify Traffic-Control Profiles That Are Currently Applied to Subscribers | 55

You use traffic-control profiles to configure traffic shaping and scheduling properties.

You can choose to configure static values or dynamic variables for the shaping parameters. The values for the dynamic variables are obtained from RADIUS when a subscriber logs in or when a subscriber changes services.

You cannot configure a traffic-control profile that contains a combination of static and dynamic parameters.

This topic includes the following tasks:

Configuring Static Traffic Shaping and Scheduling Parameters in a Dynamic Profile

To configure static traffic shaping and scheduling parameters in a traffic-control profile:

1. Create the traffic-control profile and assign a name.

```
[edit dynamic-profiles business-profile class-of-service]
user@host# edit traffic-control-profiles profile-name
```

2. Apply a static scheduler map that has been configured in the `[edit class-of-service]` hierarchy.

```
[edit dynamic-profiles business-profile class-of-service traffic-control-profiles profile-
name]
user@host# set scheduler-map map-name
```

3. Configure the shaping rate to be used in the dynamic profile.

```
[edit dynamic-profiles business-profile class-of-service traffic-control-profiles profile-
name]
user@host# set shaping-rate (rate <burst-size  bytes>
```

4. Configure the guaranteed rate to be used in the dynamic profile.

```
[edit dynamic-profiles business-profile class-of-service traffic-control-profiles profile-
name]
user@host# set guaranteed-rate (rate <burst-size  bytes>
```

5. Configure the delay-buffer rate.

   If you do not include this statement, the delay-buffer rate is based on the guaranteed rate if one is configured, or on the shaping rate if no guaranteed rate is configured.

```
[edit dynamic-profiles business-profile class-of-service traffic-control-profiles profile-
name]
user@host# set delay-buffer-rate (percent percentage | rate)
```

SEE ALSO

Configuring Static Hierarchical Scheduling in a Dynamic Profile

Example: Maintaining a Constant Traffic Flow by Configuring a Static VLAN Interface with a Dynamic Profile for Subscriber Access

Example: Configuring Dynamic Hierarchical Scheduling for Subscribers

CoS for Subscriber Access Overview | 43

Configuring Dynamic Traffic Shaping and Scheduling Parameters in a Dynamic Profile

You can configure variables for the traffic shaping and scheduling parameters. The values for the parameters are dynamically obtained by RADIUS when a subscriber logs in or changes a service.

To configure dynamic traffic-control profiles in a dynamic profile:

1. Create the traffic-control profile.

```
[edit dynamic-profiles business-profile class-of-service]
user@host# edit traffic-control-profiles profile-name
```

2. Reference a dynamic scheduler map.

   The scheduler map is dynamically configured in the `[edit dynamic-profiles profile-name class-of-service scheduler-maps]` hierarchy.

```
[edit dynamic-profiles business-profile class-of-service traffic-control-profiles profile-name]
user@host# set scheduler-map $junos-cos-scheduler-map
```

3. Configure the shaping rate variable.

```
[edit dynamic-profiles business-profile class-of-service traffic-control-profiles profile-name]
user@host# set shaping-rate $junos-cos-shaping-rate <burst-size  bytes>
```

4. Configure the guaranteed rate variable.

```
[edit dynamic-profiles business-profile class-of-service traffic-control-profiles profile-name]
user@host# set guaranteed-rate $junos-cos-guaranteed-rate <burst-size [ bytes | $junos-cos-guaranteed-rate-burst]>
```

5. Configure a variable for the delay-buffer rate.

If you do not include this statement, the delay-buffer rate is based on the guaranteed rate if one is configured, or the shaping rate if no guaranteed rate is configured.

```
[edit dynamic-profiles business-profile class-of-service traffic-control-profiles profile-name]
user@host# set delay-buffer-rate $junos-cos-delay-buffer-rate
```

SEE ALSO

Guidelines for Configuring Dynamic CoS for Subscriber Access | 44

Example: Maintaining a Constant Traffic Flow by Configuring a Static VLAN Interface with a Dynamic Profile for Subscriber Access

Example: Configuring Dynamic Hierarchical Scheduling for Subscribers

CoS for Subscriber Access Overview | 43

Using the CLI to Modify Traffic-Control Profiles That Are Currently Applied to Subscribers

IN THIS SECTION

- Using the CLI to Globally Modify a Traffic-Control Profile Currently Applied to Multiple Subscribers | 56
- Using the CLI to Modify a Traffic-Control Profile for a Specific Current Subscriber | 57

Subscriber management enables you to use the CLI to modify a traffic-control profile that is currently applied to existing subscribers. This feature allows you to update subscribers who are initially assigned the default traffic-control profile, which might have limited features.

TIP: You specify the default traffic-control profile with the `predefined-variable-defaults` statement and the `cos-traffic-control-profile` variable at the `[edit dynamic-profiles profile-name class-of-service]` hierarchy level. See *Junos OS Predefined Variables* and *Configuring Predefined Dynamic Variables in Dynamic Profiles* for more information about predefined variables.

There are two methods you can use to modify a traffic-control profile that is in use—global and per-subscriber. The global method modifies the traffic-control profile for all subscribers currently using the

traffic-control profile. The per-subscriber method modifies the traffic-control profile for a particular subscriber—all other subscribers currently using the traffic-control profile remain unaffected.

The global and per-subscriber methods share the following characteristics:

- They modify traffic-control profiles that are currently applied to active subscribers.
- Neither method creates new traffic-control profiles; they modify existing traffic-control profiles that have been previously created using the `traffic-control-profiles` statement at the `[edit dynamic-profiles profile-name class-of-service]` hierarchy level.
- Modifications are transparent to the active subscribers who are using the modified profile. The modified traffic-control profile is assigned without requiring any action by the subscriber.
- Both methods are useful when updating subscribers who are initially assigned the default traffic-control profile, which might have limited features. You specify the default traffic-control profile with the `predefined-variable-defaults` statement and the `cos-traffic-control-profile` variable at the `[edit dynamic-profiles profile-name class-of-service]` hierarchy level.

NOTE: To support CLI modification of traffic-control profiles in an IPv4/IPv6 dual-stack environment, you must have the `aggregate-clients replace` statement enabled at the `[edit system services dhcp-local-server group group-name dynamic-profile profile-name]` hierarchy

This topic includes the following tasks:

Using the CLI to Globally Modify a Traffic-Control Profile Currently Applied to Multiple Subscribers

To make a global modification for all current subscribers assigned a particular traffic-control profile, you change one or more parameters for the traffic-control profile and `commit` the changes.

In this example, the statement changes the shaping rate for the existing traffic-control profile named `TCP-silver`. After the change, the new shaping rate applies to all subscribers currently using `TCP-silver`.

1. Access the traffic-control profile you want to modify.

```
[edit dynamic-profiles business-profile class-of-service]
user@host# edit traffic-control-profiles TCP-silver
```

2. Specify the parameters that you want to modify in the traffic-control profile.

```
[edit dynamic-profiles business-profile class-of-service traffic-control-profiles TCP-silver]
user@host# set shaping-rate 20m
```

3. Commit the configuration change to update the traffic-control profile. All current subscribers using `TCP-silver` now have the new `shaping-rate`.

Using the CLI to Modify a Traffic-Control Profile for a Specific Current Subscriber

To make a per-subscriber modification for a specific subscriber that is currently assigned a traffic-control profile, you specify the name of the new traffic-control profile to use.

In this example, the command replaces the existing traffic-control profile with the profile named `TCP-gold`. The new traffic-control profile applies only to the subscriber identified by session ID `2551`.

- Request that the traffic-control profile named `TCP-gold` be applied to session ID 2551.

```
user@host> request network-access aaa subscriber modify session-id 2551 junos-cos-traffic-
control-profile TCP-gold
```

The system then displays the status message, `Successful completion`, indicating that the modification is successful. The subscriber identified by session ID 2551 now uses the `TCP-gold` traffic-control profile.

SEE ALSO

Guidelines for Configuring Dynamic CoS for Subscriber Access | 44

RELATED DOCUMENTATION

Guidelines for Configuring Dynamic CoS for Subscriber Access | 44

CoS for Subscriber Access Overview | 43

Configuring Static Hierarchical Scheduling in a Dynamic Profile

Example: Maintaining a Constant Traffic Flow by Configuring a Static VLAN Interface with a Dynamic Profile for Subscriber Access

Example: Configuring Dynamic Hierarchical Scheduling for Subscribers

Verifying the Scheduling and Shaping Configuration for Subscriber Access | 72

Configuring Schedulers in a Dynamic Profile for Subscriber Access

IN THIS SECTION

- Configuring Static Schedulers in a Dynamic Profile | 58
- Configuring Dynamic Schedulers with Variables in a Dynamic Profile | 60
- Configuring a Combination of Static and Dynamic Scheduler Parameters in a Scheduler Definition | 62

You use schedulers to define the parameters of output queues. These properties include the amount of interface bandwidth assigned to the queue, the size of the memory buffer allocated for storing packets, the priority of the queue, and the tail drop profiles associated with the queue.

You can configure up to four schedulers in a dynamic profile.

Within a dynamic profile, you can choose to define schedulers with static values, dynamic variables, or a combination of static values and dynamic variables. The dynamic variables enable RADIUS to provide the value for the scheduler parameter when the subscriber logs in.

Configuring Static Schedulers in a Dynamic Profile

This topic describes how to configure schedulers with static values in a dynamic profile for subscriber access.

To configure static scheduling and queuing in a dynamic profile:

1. Configure the scheduler and queuing parameters.
   a. Specify the scheduler for which you want to configure parameters.

```
[edit dynamic-profiles profile-name class-of-service]
user@host# edit schedulers scheduler-name
```

   b. Configure the buffer size.

```
[edit dynamic-profiles profile-name class-of-service schedulers scheduler-name]
user@host# set buffer-size remainder
```

c. Configure the drop-profile map and drop profile.

```
[edit dynamic-profiles profile-name class-of-service schedulers scheduler-name]
user@host# set drop-profile-map loss-priority any protocol any drop-profile d3
```

d. Configure the priority.

```
[edit dynamic-profiles profile-name class-of-service schedulers scheduler-name]
user@host# set priority low
```

e. Configure the transmit rate.

```
[edit dynamic-profiles profile-name class-of-service schedulers scheduler-name]
user@host# set transmit-rate percent 40
```

f. Configure the excess rate.

```
[edit dynamic-profiles profile-name class-of-service schedulers scheduler-name]
user@host# set excess-rate percent 90
```

g. (Optional) Configure the priority value for the excess-rate.

```
[edit dynamic-profiles profile-name class-of-service schedulers scheduler-name]
user@host# set excess-priority high
```

2. Associate the scheduler with a scheduler map.

a. Configure the scheduler map name.

```
[edit dynamic-profiles profile-name class-of-service]
user@host# set scheduler-maps data-smap
```

b. Configure the forwarding class.

```
[edit dynamic-profiles profile-name class-of-service scheduler-maps map-name]
user@host# set forwarding-class be
```

c. Configure the scheduler.

```
[edit dynamic-profiles profile-name class-of-service scheduler-maps map-name forwarding-
class forwarding-class-name]
user@host# set scheduler be_sch
```

SEE ALSO

Guidelines for Configuring Dynamic CoS for Subscriber Access | 44

Changing CoS Services Overview | 166

Configuring Dynamic Schedulers with Variables in a Dynamic Profile

You can configure variables for the dynamic scheduler parameters. These values are dynamically obtained by RADIUS when a subscriber logs in or changes a service using a RADIUS change of authorization (CoA) message.

To configure dynamic scheduling and queuing in a dynamic profile:

1. Configure the scheduler and queuing parameters.

   a. Specify the scheduler name using a variable.

   ```
   [edit dynamic-profiles profile-name class-of-service]
   user@host# edit schedulers $junos-cos-scheduler
   ```

   b. Configure the variable for the buffer size.

   ```
   [edit dynamic-profiles profile-name class-of-service schedulers]
   user@host# set buffer-size (percent $junos-cos-scheduler-bs | temporal $junos-cos-
   scheduler-bs)
   ```

   c. Configure the variables for the drop-profile maps and the drop profile.

   ```
   [edit dynamic-profiles profile-name class-of-service schedulers]
   user@host# set drop-profile-map loss-priority low protocol any drop-profile $junos-cos-
   scheduler-low
   user@host# set drop-profile-map loss-priority medium-low protocol any drop-profile $junos-
   cos-scheduler-medium-low
   ```

```
user@host# set drop-profile-map loss-priority medium-high protocol any drop-profile $junos-
cos-scheduler-medium-high
user@host# set drop-profile-map loss-priority high protocol any drop-profile $junos-cos-
scheduler-high
user@host# set drop-profile-map loss-priority any protocol any drop-profile $junos-cos-
scheduler-any
```

d. Configure the variable for the priority.

```
[edit dynamic-profiles profile-name class-of-service schedulers]
user@host# set priority $junos-cos-scheduler-pri
```

e. Configure the variable for the transmit rate.

```
[edit dynamic-profiles profile-name class-of-service schedulers]
user@host# set transmit-rate $junos-cos-scheduler-tx
```

f. Configure the variable for the excess rate.

```
[edit dynamic-profiles profile-name class-of-service schedulers]
user@host# set excess-rate percent $junos-cos-scheduler-excess-rate
```

g. Configure the variable for the priority of the excess-rate.

```
[edit dynamic-profiles profile-name class-of-service schedulers]
user@host# set excess-priority $junos-cos-scheduler-excess-priority
```

2. Associate the scheduler with a scheduler map.

a. Configure the scheduler map name.

```
[edit dynamic-profiles profile-name class-of-service]
user@host# edit scheduler-maps scheduler-map-name
```

b. Configure the forwarding class.

```
[edit dynamic-profiles profile-name class-of-service scheduler-maps scheduler-map-name]
user@host# set forwarding-class be
```

c. Configure the scheduler.

```
[edit dynamic-profiles profile-name class-of-service scheduler-maps scheduler-map-name]
user@host# set scheduler $junos-cos-scheduler
```

SEE ALSO

Guidelines for Configuring Dynamic CoS for Subscriber Access | 44

Changing CoS Services Overview | 166

Configuring a Combination of Static and Dynamic Scheduler Parameters in a Scheduler Definition

Within a dynamic profile, you can choose to configure one dynamic scheduler definition, or combine static and dynamic scheduler parameters in many static scheduler definitions.

Combining static and dynamic scheduler parameters enables you to provide subscribers with unique rate configurations that the RADIUS definitions for predefined variables do not allow.

To configure a scheduler definition that contains static and dynamic scheduling and queuing parameters:

1. Configure the scheduler definition.

   a. Specify the scheduler name.

   NOTE: To configure a static scheduler that contains both static and dynamic parameters, you must specify a unique scheduler name, not the `$junos-cos-scheduler` variable.

```
[edit dynamic-profiles profile-name class-of-service]
user@host# edit schedulers scheduler-name
```

   b. Configure the buffer size.

   Do either of the following:

- Configure a static value.

```
[edit dynamic-profiles profile-name class-of-service schedulers scheduler-name]
user@host# set buffer-size (percent percentage | remainder | temporal (microseconds)
```

- Configure a variable.

```
[edit dynamic-profiles profile-name class-of-service schedulers scheduler-name]
user@host# set buffer-size (percent $junos-cos-scheduler-bs | temporal $junos-cos-
scheduler-bs)
```

c. Configure the drop-profile maps, the drop profile, and the priority.

Do either of the following:

- Configure static values.

```
[edit dynamic-profiles profile-name class-of-service schedulers scheduler-name]
user@host# set drop-profile-map loss-priority any protocol any drop-profile d3
```

```
[edit dynamic-profiles profile-name class-of-service schedulers scheduler-name]
user@host# set priority low
```

- Configure variables.

```
[edit dynamic-profiles profile-name class-of-service schedulers scheduler-name]
user@host# set drop-profile-map loss-priority low protocol any drop-profile $junos-cos-
scheduler-low
user@host# set drop-profile-map loss-priority medium-low protocol any drop-profile
$junos-cos-scheduler-medium-low
user@host# set drop-profile-map loss-priority medium-high protocol any drop-profile
$junos-cos-scheduler-medium-high
user@host# set drop-profile-map loss-priority high protocol any drop-profile $junos-
cos-scheduler-high
user@host# set drop-profile-map loss-priority any protocol any drop-profile $junos-cos-
scheduler-any
```

d. Configure the priority.

Do either of the following:

- Configure a static value.

```
[edit dynamic-profiles profile-name class-of-service schedulers scheduler-name]
user@host# set excess-priority high
```

- Configure a variable.

```
[edit dynamic-profiles profile-name class-of-service schedulers scheduler-name]
user@host# set excess-priority $junos-cos-scheduler-excess-priority
```

e. Configure the transmit rate.

Do either of the following:

- Configure a static value.

```
[edit dynamic-profiles profile-name class-of-service schedulers scheduler-name]
user@host# set transmit-rate
```

- Configure a variable.

```
[edit dynamic-profiles profile-name class-of-service schedulers scheduler-name]
user@host# set transmit-rate $junos-cos-scheduler-tx
```

f. Configure the excess rate.

Do either of the following:

- Configure a static value.

```
[edit dynamic-profiles profile-name class-of-service schedulers scheduler-name]
user@host# set excess-rate percent 250
```

- Configure a variable.

```
[edit dynamic-profiles profile-name class-of-service schedulers scheduler-name]
user@host# set excess-rate percent $junos-cos-scheduler-excess-rate
```

g. Configure the priority for the excess-rate.

Do either of the following:

- Configure a static value.

```
[edit dynamic-profiles profile-name class-of-service schedulers scheduler-name]
user@host# set excess-priority high
```

- Configure a variable.

```
[edit dynamic-profiles profile-name class-of-service schedulers scheduler-name]
user@host# set excess-priority percent $junos-cos-scheduler-excess-priority
```

2. Associate the scheduler with a scheduler map.

a. Configure the scheduler map name.

```
[edit dynamic-profiles profile-name class-of-service]
user@host# edit scheduler-maps scheduler-map-name
```

b. Configure the forwarding class.

```
[edit dynamic-profiles profile-name class-of-service scheduler-maps scheduler-map-name]
user@host# set forwarding-class be
```

c. Configure the scheduler.

```
[edit dynamic-profiles profile-name class-of-service scheduler-maps scheduler-map-name]
user@host# set scheduler $junos-cos-scheduler
```

SEE ALSO

Guidelines for Configuring Dynamic CoS for Subscriber Access | 44

Changing CoS Services Overview | 166

RELATED DOCUMENTATION

Guidelines for Configuring Dynamic CoS for Subscriber Access | 44

Verifying the Scheduling and Shaping Configuration for Subscriber Access | 72

Changing CoS Services Overview | 166

Configuring Scheduler and Scheduler Map Sharing

The system generates unique identifiers (IDs) in dynamic profiles created for services. The generated unique IDs enable you to identify and configure separate parameter values with the same variable name. When applied to CoS, you can configure scheduler and scheduler map sharing. In client-access profiles, schedulers and scheduler maps must use the unique ID format. If the client-access profile uses the unique ID format and you want to have either scheduler or scheduler map sharing for service activation, you must configure the service profile in unique ID format. Generating unique IDs based on schedulers and scheduler maps eliminates duplication and improves router performance and scalability. You can configure scheduler and scheduler map sharing by including the variables for CoS in the client access or service dynamic profile. All scheduler maps and schedulers must be in the unique ID format.

Before you configure variables for the client access or service dynamic profile:

- Create a basic dynamic profile.

  See *Configuring a Basic Dynamic Profile*.

To configure variables for the client access or service dynamic profile:

1. Configure the variables for the dynamic client access profile.

```
[edit dynamic-profiles client-profile variables]
user@host# set smap_data uid
user@host# set data_sched uid
```

2. Configure the CoS parameters for the variables in the scheduler profile.

```
[edit dynamic-profiles client-profile class-of-service]
user@host# edit schedulers “$data_sched”
user@host# set transmit-rate percent 10
user@host# set buffer-size remainder
user@host# set priority low
```

3. Configure the CoS parameters for the variables in the scheduler maps profile.

```
[edit dynamic-profiles client-profile class-of-service]
user@host# edit scheduler-maps “$smap_data”
user@host# edit  forwarding-class be scheduler “$data_sched”
```

For example, you can configure scheduler maps and schedulers for a client access profile:

```
dynamic-profiles {
    cos-para {
        variables {
            data_smap uid;
            data_video_smap uid;
            voice_smap uid;
            data_sched uid;
            video_sched uid;
            voice_sched uid;
        }
        …
        class-of-service {
            traffic-control-profiles {
                tcp1 {
                    scheduler-map "$junos-cos-scheduler-map";
                    shaping-rate "$junos-cos-shaping-rate";
                    guaranteed-rate 10m;
                    delay-buffer-rate "$junos-cos-delay-buffer-rate";
                }
            }
            interfaces {
                "$junos-interface-ifd-name" {
                    unit "$junos-underlying-interface-unit" {
                        output-traffic-control-profile tcp1;
                    }
                }
            }
            scheduler-maps {
                "$data_smap" {
                    forwarding-class be scheduler "$data_sched";
                }
                "$data_video_smap" {
                    forwarding-class be scheduler "$data_sched";
```

```
                    forwarding-class af scheduler "$video_sched";
                }
                “$voice_smap” {
                    forwarding-class ef scheduler “$voice_sched”;
                }
            }
            schedulers {
                "$data_sched" {
                    transmit-rate "$junos-cos-scheduler-tx";
                    inactive: buffer-size percent "$junos-cos-scheduler-bs";
                    priority "$junos-cos-scheduler-pri";
                }
                "$video_sched" {
                    transmit-rate "$junos-cos-scheduler-tx";
                    inactive: buffer-size percent "$junos-cos-scheduler-bs";
                    priority "$junos-cos-scheduler-pri";
                }
                “$voice_sched” {
                    transmit-rate percent 10;
                    buffer-size remainder;;
                    priority low;
                }
            }
        }
    }
}
```

RELATED DOCUMENTATION

Dynamic Profiles Overview

Guidelines for Configuring Dynamic CoS for Subscriber Access | 44

Example: Providing Unique Rate Configurations for Schedulers in a Dynamic Profile

Combining static and dynamic schedulers in a dynamic profile enables you to provide subscribers with services that have unique scheduler definitions.

In this example, the network administrator configures the data service with a `transmit-rate` that is rate controlled using the `$junos-cos-scheduler-tx` predefined variable. RADIUS dynamically supplies the percentage value for the transmission rate that is specified in the RADIUS VSA to the data scheduler when the subscriber logs in.

For the best-effort service, the network administrator assigns the remaining transmission rate that is available.

```
schedulers {
    data-scheduler {
        transmit-rate percent rate-limit $junos-cos-scheduler-tx;
        buffer-size percent $junos-cos-scheduler-bs;
        priority $junos-cos-scheduler-pri;
        drop-profile-map loss-priority low protocol any drop-profile d0;
        drop-profile-map loss-priority medium-low protocol any drop-profile d1;
        drop-profile-map loss-priority medium-high protocol any drop-profile d2;
        drop-profile-map loss-priority high protocol any drop-profile d3;
        drop-profile-map loss-priority any protocol any drop-profile all;
    }
    best-effort-scheduler {
        transmit-rate remainder;
        buffer-size percent $junos-cos-scheduler-bs;
        priority medium-high;
        drop-profile-map loss-priority low protocol any drop-profile $junos-cos-scheduler-
dropfile-low;
        drop-profile-map loss-priority medium-low protocol any drop-profile d1;
        drop-profile-map loss-priority medium-high protocol any drop-profile $junos-cos-
scheduler-dropfile-medium-high;
        drop-profile-map loss-priority high protocol any drop-profile d3;
        drop-profile-map loss-priority any protocol any drop-profile $junos-cos-scheduler-
dropfile-any;
    }
```

RELATED DOCUMENTATION

Configuring Schedulers in a Dynamic Profile for Subscriber Access | 58

Example: Configuring Aggregate Scheduling of Queues for Residential Subscribers on Static IP Demux Interfaces

In this example, scheduling is configured for a residential subscriber. Each forwarding class represents a multiplay service (voice, video, and data), and is equivalent to a queue.

An interface set of IP demux interfaces represents a DSLAM, and provides shaping of subscribers services to a DSLAM aggregate rate.

```
[edit]
interfaces {
    interface-set demux-set {
        interface demux0 {
            unit 0;
            unit 1;
        }
    }
    ge-2/0/1 {
        vlan-tagging;
        unit 1 {
            per-session-scheduler;
            vlan-id 1;
            demux-source inet;
            family inet {
                address 192.0.2.4/24;
            }
        }
    }
    demux0 {
        unit 0 {
            demux-options {
                underlying-interface ge-2/0/1.1;
            }
            family inet {
                address 192.0.2.1/24;
                demux-source {
                    192.0.2.0/24;
                }
            }
        }
        unit 1 {
```

```
            demux-options {
                underlying-interface ge-2/0/1.1;
            }
            family inet {
                address 192.0.2.21/24;
                demux-source {
                    192.0.2.20/24;
                }
            }
        }
    }
}
class-of-service {
    traffic-control-profiles {
        T1 {
            scheduler-map m1;
            shaping-rate 5m;
        }
        T2 {
            shaping-rate 60m;
        }
    }
    interfaces {
        interface-set demux-set {
            output-traffic-control-profile T2;
        }
        demux0 {
            unit 0 {
                output-traffic-control-profile T1;
            }
            unit 1 {
                output-traffic-control-profile T1;
            }
        }
    }
    scheduler-maps {
        m1 {
            forwarding-class best-effort scheduler s0;
            forwarding-class expedited-forwarding scheduler s1;
            forwarding-class assured-forwarding scheduler s2;
            forwarding-class network-control scheduler s3;
        }
    }
```

```
    schedulers {
        s0 {
            transmit-rate percent 10;
            buffer-size percent 10;
        }
        s1 {
            transmit-rate percent 20;
            buffer-size percent 20;
        }
        s2 {
            transmit-rate percent 30;
            buffer-size percent 30;
        }
        s3 {
            transmit-rate percent 40;
            buffer-size percent 40;
        }
    }
}
```

Verifying the Scheduling and Shaping Configuration for Subscriber Access

IN THIS SECTION

- Purpose | 72
- Action | 73

Purpose

View the class-of-service (CoS) configurations that are referenced in a dynamic profile for subscriber access.

Action

- To display the entire CoS configuration, including static and dynamic parameters:

```
user@host>  show class-of-service
```

- To display the CoS configuration for a subscriber interface:

```
user@host> show class-of-service interface
```

- To display traffic shaping and scheduling profiles:

```
user@host> show class-of-service traffic-control-profile
```

- To display the mapping of schedulers to forwarding classes and a summary of scheduler parameters for each entry:

```
user@host>  show class-of-service scheduler-map
```

CHAPTER 4

Allocating Dedicated Queues for Each Logical Interface Using Per-Unit Scheduling

IN THIS CHAPTER

- Hardware Requirements for Dynamic Per-Unit Scheduling | 74
- Configuring Per-Unit Scheduling in a Dynamic Profile | 75
- Example: Configuring Per-Unit Scheduling for Subscriber Access | 77

Hardware Requirements for Dynamic Per-Unit Scheduling

Table 12 on page 74 lists the hardware requirements based on subscriber interface type for per-unit scheduling in dynamic CoS configurations.

Table 12: Hardware Required for Per-Unit Scheduling Dynamic CoS Configurations

Subscriber Interface Type	EQ DPCs on MX Series Routers	MPC/MIC Modules on MX Series Routers
Static and dynamic VLANs	Yes	Yes
Static and dynamic VLANs over aggregated Ethernet	No	No
Static or dynamic IP demux interfaces	Yes	No
Static or dynamic IP demux interfaces over aggregated Ethernet	No	No

Table 12: Hardware Required for Per-Unit Scheduling Dynamic CoS Configurations *(Continued)*

Subscriber Interface Type	EQ DPCs on MX Series Routers	MPC/MIC Modules on MX Series Routers
Static or dynamic VLAN demux interfaces	No	No
Static or dynamic VLAN demux interfaces over aggregated Ethernet	No	No
Static PPPoE interfaces	No	Yes
Dynamic PPPoE interfaces	No	No
Static or dynamic PPPoE interfaces over aggregated Ethernet	No	No
L2TP LAC tunnel over PPP	No	No
L2TP LNS inline service over PPP	No	No

RELATED DOCUMENTATION

Guidelines for Configuring Dynamic CoS for Subscriber Access | 44

Configuring Per-Unit Scheduling in a Dynamic Profile | 75

Configuring Per-Unit Scheduling in a Dynamic Profile

Per-unit scheduling enables one set of output queues for each logical interface configured under the physical interface. In per-unit scheduling configurations, each Layer 3 scheduler node is allocated a dedicated set of queues.

If you do not explicitly configure CoS parameters, a default traffic profile with queues is attached to the logical interface.

To configure per-unit scheduling and queuing for subscriber access:

1. Configure the static CoS parameters in the `[edit class-of-service]` hierarchy.

   a. Enable the per-unit scheduler for the physical interface.

   ```
   [edit interfaces interface-name]
   user@host# set per-unit-scheduler
   ```

   b. Configure the drop profiles.

   See Defining Packet Drop Behavior by Configuring RED Drop Profiles.

   c. Configure the forwarding classes.

   See Configuring a Custom Forwarding Class for Each Queue.

   d. Configure the rewrite-rules and classifier definitions.

   See Configuring Rewrite Rules and Configuring Behavior Aggregate Classifiers.

   See The Junos OS CoS Components Used to Manage Congestion and Control Service Levels for information about configuring the remaining CoS parameters.

2. Configure a static or dynamic subscriber interface that can be referenced in the dynamic profile.
3. Configure CoS parameters in a dynamic profile.

   a. Configure the dynamic profile.

   See *Configuring a Basic Dynamic Profile*.

   b. Configure traffic shaping and scheduling parameters in the dynamic profile using a traffic-control profile.

   See "Configuring Traffic Scheduling and Shaping for Subscriber Access" on page 52.

   c. Configure the schedulers and scheduler map in the dynamic profile.

   You can configure the schedulers using dynamic variables or a combination of both static values and dynamic variables.

   See "Configuring Schedulers in a Dynamic Profile for Subscriber Access" on page 58.

   d. Apply CoS parameters to a subscriber interface by referencing an interface in the dynamic profile.

   - For traffic shaping and scheduling, see "Applying Traffic Shaping and Scheduling to a Subscriber Interface in a Dynamic Profile" on page 228.
   - For rewrite rules, see "Applying a Rewrite Rule Definition to a Subscriber Interface in a Dynamic Profile" on page 231.
   - For classifiers, see "Applying a Classifier to a Subscriber Interface in a Dynamic Profile" on page 232.

4. (Optional) Configure variables in access and service profiles to enable RADIUS to activate subscriber and upgrade services through CoA.

   NOTE: Do not instantiate a CoA request using a service dynamic profile that is already in use on the same logical interface.

   Because you have configured the scheduler map in the dynamic profile, queues are merged when subscribers change services. Other CoS parameters are replaced.

   When multiple subscribers are enabled on a DHCP subscriber interface, and the dynamic profile referenced by DHCP does not have the `replace` keyword configured, the system does not replace the parameters. Instead, it combines the values of the parameters to their maximum scalar value.

   a. Configure CoS variables in a dynamic profile.

      See "Configuring Static Default Values for Traffic Scheduling and Shaping" on page 174

   b. (Optional) Enable multiple clients for the same subscriber (logical interface) to aggregate attributes by configuring the `aggregate-clients` option for the dynamic profile attached to a DHCP subscriber interface.

      See *Attaching Dynamic Profiles to DHCP Subscriber Interfaces or DHCP Client Interfaces*.

RELATED DOCUMENTATION

CoS for Subscriber Access Overview | 43

Guidelines for Configuring Dynamic CoS for Subscriber Access | 44

Example: Configuring Per-Unit Scheduling for Subscriber Access | 77

Example: Configuring Per-Unit Scheduling for Subscriber Access

In this example, a network administrator sets up a subscriber access configuration with per-unit scheduling.

1. The administrator configures the static VLAN interfaces and enables per-unit scheduling for the interfaces.

```
[edit]
    interfaces {
    ge-1/1/0 {
```

```
        per-unit-scheduler;
        vlan-tagging;
        unit 100 {
            vlan-id 100;
            family inet {
                unnumbered-address lo0.0 preferred-source-address 192.0.2.100;
            }
        }
        unit 200 {
            vlan-id 200;
            family inet {
                unnumbered-address lo0.0 preferred-source-address 192.0.2.100;
            }
        }
    }
    ge-1/1/1 {
        per-unit-scheduler;
        vlan-tagging;
        unit 100 {
            vlan-id 100;
            family inet {
                unnumbered-address lo0.0 preferred-source-address 192.0.2.100;
            }
        }
        unit 200 {
            vlan-id 200;
            family inet {
                unnumbered-address lo0.0 preferred-source-address 192.0.2.100;
            }
        }
    }
    ge-1/0/1 {
        unit 0 {
            family inet {
                address 203.0.113.31/24;
            }
        }
    }
    ge-1/1/2 {
        description "wfce14 eth1 soso ge-1/1/2";
        vlan-tagging;
        gigether-options {
            no-auto-negotiation;
```

```
        }
        unit 100 {
            vlan-id 100;
            family inet {
                address 203.0.113.121/24;
            }
        }
    }
}
```

2. The administrator configures static CoS parameters, including forwarding classes and classifiers, to be referenced in the dynamic profiles.

```
[edit]
    class-of-service {
    classifiers {
        inet-precedence 8q-inet {
            forwarding-class be {
                loss-priority low code-points 000;
            }
            forwarding-class ef {
                loss-priority low code-points 001;
            }
            forwarding-class af {
                loss-priority low code-points 010;
            }
            forwarding-class nc {
                loss-priority low code-points 011;
            }
            forwarding-class voice {
                loss-priority low code-points 100;
            }
            forwarding-class video {
                loss-priority low code-points 101;
            }
            forwarding-class game {
                loss-priority low code-points 110;
            }
            forwarding-class data {
                loss-priority low code-points 111;
            }
        }
```

```
        inet-precedence 4q-inet {
            forwarding-class be {
                loss-priority low code-points [ 000 001 ];
            }
            forwarding-class ef {
                loss-priority low code-points [ 010 011 ];
            }
            forwarding-class af {
                loss-priority low code-points [ 100 101 ];
            }
            forwarding-class nc {
                loss-priority low code-points [ 110 111 ];
            }
        }
        inet-precedence 8q-drop-inet {
            forwarding-class be {
                loss-priority low code-points 000;
            }
            forwarding-class ef {
                loss-priority medium-low code-points 001;
            }
            forwarding-class af {
                loss-priority medium-high code-points 010;
            }
            forwarding-class nc {
                loss-priority high code-points 011;
            }
            forwarding-class voice {
                loss-priority low code-points 100;
            }
            forwarding-class video {
                loss-priority medium-low code-points 101;
            }
            forwarding-class game {
                loss-priority medium-high code-points 110;
            }
            forwarding-class data {
                loss-priority high code-points 111;
            }
        }
        inet-precedence 4q-drop-inet {
            forwarding-class be {
                loss-priority low code-points [ 000 001 ];
```

```
            }
            forwarding-class ef {
                loss-priority medium-low code-points [ 010 011 ];
            }
            forwarding-class af {
                loss-priority medium-high code-points [ 100 101 ];
            }
            forwarding-class nc {
                loss-priority high code-points [ 110 111 ];
            }
        }

    }
    drop-profiles {
        d0 {
            fill-level 25 drop-probability 100;
            fill-level 0 drop-probability 0;
        }
        d1 {
            fill-level 50 drop-probability 100;
            fill-level 0 drop-probability 0;
        }
        d2 {
            fill-level 75 drop-probability 100;
            fill-level 0 drop-probability 0;
        }
        d3 {
            fill-level 100 drop-probability 100;
            fill-level 0 drop-probability 0;
        }
        all {
            fill-level 0 drop-probability 0;
            fill-level 100 drop-probability 100;
        }
    }
    forwarding-classes {
        queue 0 be;
        queue 1 ef;
        queue 2 af;
        queue 3 nc;
        queue 4 voice;
        queue 5 video;
        queue 6 game;
```

```
        queue 7 data;
    }

    interfaces {
        ge-1/0/1 {
            unit 0 {
                classifiers {
                    inet-precedence 8q-drop-low-high-inet;
                }
            }
        }
    }
    traceoptions {
        flag all;
        flag asynch;
        flag route-socket;
    }
}
```

3. The administrator configures the access and service dynamic profiles to receive CoS parameters for the subscriber interfaces through RADIUS.

```
[edit]
    dynamic-profiles {
    subscriber {
        interfaces {
            "$junos-interface-ifd-name" {
                unit "$junos-underlying-interface-unit" {
                    family inet;
                }
            }
        }
        class-of-service {
            traffic-control-profiles {
                zero {
                    scheduler-map "$junos-cos-scheduler-map";
                    shaping-rate "$junos-cos-shaping-rate";
                    guaranteed-rate "$junos-cos-guaranteed-rate";
                    delay-buffer-rate "$junos-cos-delay-buffer-rate";
                }
            }
            interfaces {
```

```
            "$junos-interface-ifd-name" {
                unit "$junos-underlying-interface-unit" {
                    output-traffic-control-profile zero;
                }
            }
        }
        scheduler-maps {
            be_smap {
                forwarding-class be scheduler be_sch;
            }
            all_smap {
                forwarding-class be scheduler be_sch;
                forwarding-class ef scheduler ef_sch;
                forwarding-class af scheduler af_sch;
                forwarding-class nc scheduler nc_sch;
                forwarding-class video scheduler video_sch;
                forwarding-class data scheduler data_sch;
            }
            be_ef_smap {
                forwarding-class be scheduler be_sch;
                forwarding-class ef scheduler ef_sch;
            }
            af_smap {
                forwarding-class af scheduler af_sch;
            }
            be_ef_af_nc_smap {
                forwarding-class be scheduler be_sch;
                forwarding-class ef scheduler ef_sch;
                forwarding-class af scheduler af_sch;
                forwarding-class nc scheduler nc_sch;
            }
            voice_video_game_data_smap {
                forwarding-class voice scheduler voice_sch;
                forwarding-class video scheduler video_sch;
                forwarding-class game scheduler game_sch;
                forwarding-class data scheduler data_sch;
            }
        }
        schedulers {
            "$junos-cos-scheduler" {
                transmit-rate percent "$junos-cos-scheduler-tx";
                buffer-size percent "$junos-cos-scheduler-bs";
                priority "$junos-cos-scheduler-pri";
```

```
                    drop-profile-map loss-priority low protocol any drop-profile "$junos-cos-
scheduler-dropfile-low";
                    drop-profile-map loss-priority medium-low protocol any drop-profile
"$junos-cos-scheduler-dropfile-medium-low";
                    drop-profile-map loss-priority medium-high protocol any drop-profile
"$junos-cos-scheduler-dropfile-medium-high";
                    drop-profile-map loss-priority high protocol any drop-profile "$junos-cos-
scheduler-dropfile-high";
                }
            }
        }
    }
    service {
        variables {
            fc_1 default-value be;
            sch_1 default-value be_sch;
            sch-tx_1 default-value 20000000;
            sch-bs_1 default-value 10;
            sch-pri_1 default-value high;
            sch-drop-low_1 default-value d3;
            sch-drop-med-low_1 default-value d2;
            sch-drop-med-high_1 default-value d1;
            sch-drop-high_1 default-value d0;
            sch-drop-any_1 default-value d3;
            fc_2 default-value af;
            sch_2 default-value af_sch;
            sch-tx_2 default-value 10;
            sch-bs_2 default-value 10;
            sch-pri_2 default-value high;
            sch-drop-low_2 default-value d3;
            sch-drop-med-low_2 default-value d2;
            sch-drop-med-high_2 default-value d1;
            sch-drop-high_2 default-value d0;
            sch-drop-any_2 default-value d3;
            fc_3 default-value voice;
            sch_3 default-value voice_sch;
            sch-tx_3 default-value 20000000;
            sch-bs_3 default-value 10;
            sch-pri_3 default-value high;
            sch-drop-low_3 default-value d3;
            sch-drop-med-low_3 default-value d2;
            sch-drop-med-high_3 default-value d1;
            sch-drop-high_3 default-value d0;
```

```
            sch-drop-any_3 default-value d3;
            fc_4 default-value game;
            sch_4 default-value game_sch;
            sch-tx_4 default-value 10;
            sch-bs_4 default-value 10;
            sch-pri_4 default-value high;
            sch-drop-low_4 default-value d3;
            sch-drop-med-low_4 default-value d2;
            sch-drop-med-high_4 default-value d1;
            sch-drop-high_4 default-value d0;
            sch-drop-any_4 default-value d3;
            scheduler-map default-value all_smap;
        }
        class-of-service {
            scheduler-maps {
                "$scheduler-map" {
                    forwarding-class "$fc_1" scheduler "$sch_1";
                    forwarding-class "$fc_2" scheduler "$sch_2";
                    forwarding-class "$fc_3" scheduler "$sch_3";
                    forwarding-class "$fc_4" scheduler "$sch_4";
                }
            }
            schedulers {
                "$sch_1" {
                    transmit-rate "$sch-tx_1";
                    buffer-size percent "$sch-bs_1";
                    priority "$sch-pri_1";
                    drop-profile-map loss-priority low protocol any drop-profile "$sch-drop-
low_1";
                    drop-profile-map loss-priority medium-low protocol any drop-profile "$sch-
drop-med-low_1";
                    drop-profile-map loss-priority medium-high protocol any drop-profile
"$sch-drop-med-high_1";
                    drop-profile-map loss-priority high protocol any drop-profile "$sch-drop-
high_1";
                }
                "$sch_2" {
                    transmit-rate percent "$sch-tx_2";
                    buffer-size percent "$sch-bs_2";
                    priority "$sch-pri_2";
                    drop-profile-map loss-priority low protocol any drop-profile "$sch-drop-
low_2";
                    drop-profile-map loss-priority medium-low protocol any drop-profile "$sch-
```

```
drop-med-low_2";
                    drop-profile-map loss-priority medium-high protocol any drop-profile
"$sch-drop-med-high_2";
                    drop-profile-map loss-priority high protocol any drop-profile "$sch-drop-
high_2";
                }
                "$sch_3" {
                    transmit-rate "$sch-tx_3";
                    buffer-size percent "$sch-bs_3";
                    priority "$sch-pri_3";
                    drop-profile-map loss-priority low protocol any drop-profile "$sch-drop-
low_3";
                    drop-profile-map loss-priority medium-low protocol any drop-profile "$sch-
drop-med-low_3";
                    drop-profile-map loss-priority medium-high protocol any drop-profile
"$sch-drop-med-high_3";
                    drop-profile-map loss-priority high protocol any drop-profile "$sch-drop-
high_3";
                }
                "$sch_4" {
                    transmit-rate percent "$sch-tx_4";
                    buffer-size percent "$sch-bs_4";
                    priority "$sch-pri_4";
                    drop-profile-map loss-priority low protocol any drop-profile "$sch-drop-
low_4";
                    drop-profile-map loss-priority medium-low protocol any drop-profile "$sch-
drop-med-low_4";
                    drop-profile-map loss-priority medium-high protocol any drop-profile
"$sch-drop-med-high_4";
                    drop-profile-map loss-priority high protocol any drop-profile "$sch-drop-
high_4";
                }
            }
        }
    }
    service_2 {
        variables {
            fc_1 default-value be;
            sch_1 default-value be_sch;
            sch-tx_1 default-value 10;
            sch-bs_1 default-value 10;
            sch-pri_1 default-value high;
            sch-drop-low_1 default-value d3;
```

```
            sch-drop-med-low_1 default-value d2;
            sch-drop-med-high_1 default-value d1;
            sch-drop-high_1 default-value d0;
            sch-drop-any_1 default-value d3;
            scheduler-map default-value all_smap;
        }
        class-of-service {
            scheduler-maps {
                "$scheduler-map" {
                    forwarding-class "$fc_1" scheduler "$sch_1";
                }
            }
            schedulers {
                "$sch_1" {
                    transmit-rate percent "$sch-tx_1";
                    buffer-size percent "$sch-bs_1";
                    priority "$sch-pri_1";
                    drop-profile-map loss-priority low protocol any drop-profile "$sch-drop-
low_1";
                    drop-profile-map loss-priority medium-low protocol any drop-profile "$sch-
drop-med-low_1";
                    drop-profile-map loss-priority medium-high protocol any drop-profile
"$sch-drop-med-high_1";
                    drop-profile-map loss-priority high protocol any drop-profile "$sch-drop-
high_1";
                }
            }
        }
    }
}
```

4. The network administrator configures DHCP and RADIUS to grant access and services to the interfaces referenced by the `subscriber` dynamic profile.

```
[edit]
    forwarding-options {
    dhcp-relay {
        traceoptions {
            file size 1g;
            flag all;
        }
        dynamic-profile subscriber aggregate-clients replace;
```

```
        server-group {
            subscriber-server {
                203.0.113.2;
            }
        }
        active-server-group subscriber-server;
        group relay-0 {
            authentication {
                password $ABC123;
                username-include {
                    user-prefix user0;
                    mac-address;
                }
            }
            interface ge-1/1/0.100;
            interface ge-1/1/0.200;
        }
    }
}
radius-server {
    198.51.100.11 secret "$ABC123$ABC123$ABC123"; ## SECRET-DATA
}
    profile subscriber-profile {
        authentication-order radius;
        radius {
            authentication-server 198.51.100.11;
            accounting-server 198.51.100.11;
        }
        radius-server {
            198.51.100.11 secret "$ABC123$ABC123"; ## SECRET-DATA
        }
        accounting {
            order radius;
            statistics time;
        }
    }
```

RELATED DOCUMENTATION

Configuring Per-Unit Scheduling in a Dynamic Profile | 75

CHAPTER 5

Configuring Hierarchical CoS Scheduling on MPLS Ethernet Pseudowire Subscriber Interfaces

IN THIS CHAPTER

- Enhanced Subscriber Management Subscriber Logical Interfaces or Interface Sets Over Underlying Logical Interfaces for a CoS scheduler Hierarchy | **89**
- Enhanced Subscriber Management Subscriber Logical Interfaces or Interface Sets Over MPLS Pseudowires for a CoS scheduler Hierarchy | **92**
- Configuring Layer 2 Subscriber Logical Interfaces for CoS Hierarchical Schedulers Using Dynamic Profiles for Differentiating Home and Access Node Networks | **95**
- Example: Configuring Layer 2 Subscriber Logical Interfaces for CoS Hierarchical Schedulers Using Static CoS for Differentiating Home and Access Node Networks | **101**

Enhanced Subscriber Management Subscriber Logical Interfaces or Interface Sets Over Underlying Logical Interfaces for a CoS scheduler Hierarchy

Starting in Junos OS Release 15.1, you can enable a CoS scheduling hierarchy for subscriber logical interfaces or interface sets over underlying logical interfaces. Until Junos OS Release 14.2, an interface set can be either at Layer 2 or Layer 3 levels of the CoS three-level hierarchical scheduler. When the interface set is at the Layer 3 level, a mechanism to configure the Layer 2 node to which the Layer 3 node belonged was not available. As a result, the Layer 2 node was a dummy node in such a case for the three-level hierarchical scheduler.

In certain Broadband Remote Access Server (B-RAS) deployments, when you use an interface set to denote a home network, it might be necessary to configure the home network and the access node (such as a digital subscriber line access multiplexer, or DSLAM) in a scheduler hierarchy. This method of hierarchical scheduler is necessary in agent circuit identifier (ACI) VLANs because a home or an ACI is always an interface set in such topologies. You can now enable an enhanced subscriber management logical interface, such as an MPLS pseudowire logical interface to function as a Layer 2 node in a CoS hierarchical scheduler. A subscriber logical interface is considered to operate at Layer 2 only if you configure three-level hierarchical scheduling on the logical tunnel anchor point on the physical interface

(the IFD). An MPLS pseudowire is a virtual device that is stacked above the logical tunnel anchor point. Implicit hierarchy processes the interface stack properly in such a setup. To configure three-level hierarchical scheduling, include the `implicit- hierarchy` option at the `[edit interfaces "$junos-interface-ifd-name" hierarchical-scheduler]` or the `[edit interfaces lt-device hierarchical-scheduler]` hierarchy level. If the `implicit-hierarchy` option is not set on the logical tunnel anchor point, logical interfaces behave normally with the hierarchical-scheduler mode configured with or without the `hierarchical-scheduler maximum-hierarchy-levels` option under the `[edit interfaces` *`interface-name`* `hierarchical-scheduler]` statement.

In this case, when you apply a traffic-control profile to the pseudowire and service logical interfaces, they both reside in level 3 scheduler nodes and do not form a scheduling hierarchy, which might not be the desirable behavior. Subscriber logical interfaces at Layer 3 that are stacked over the underlying logical interfaces at Layer 2 are supported if the Layer 2 logical interface is an underlying interface of the Layer 3 interface.

For example, if a PPPoE logical interface contains an underlying logical interface, ge-1/0/0.100, the ge-1/0/0.100 interface can be at Layer 2 and the PPPoE logical interface can be at Layer 3. You can also configure PPP or IP demux interfaces in such a fashion at Layer 3. Similarly, you can configure logical interfaces at Layer 2 that serve as underlying interfaces for logical interface sets, such as PPPoE ACI interface sets or IP demux interface sets, where all the member logical interfaces of the interface set contain the same underlying logical interface at Layer 2. You can configure the logical interfaces at Layer 2 in a dynamic profile or in a static CoS configuration.

Dynamic profile CoS configuration for underlying logical interfaces is supported because two interface stanzas with TCPs in one dynamic profile are considered valid. For dynamic underlying logical interfaces, you can configure in a profile different from the client logical interface profile or in the same profile as the client profile. If the underlying logical interface is static and CoS is configured dynamically in a dynamic profile, it must be specified in the same profile as the client logical interface. However, CoS for the underlying logical interfaces must be configured either in a dynamic profile or in a static CoS because both static CoS and dynamic CoS are not supported on the same logical interface.

Reparenting is a technique that denotes the movement of the CoS hierarchical scheduler from one node to another node, such as moving all logical interfaces stacked over an underlying logical interface on top of the base physical interface to be over the underlying logical interface directly and adding the scheduling node. This movement might occur when when CoS for the underlying logical interface or the underlying interface set is configured later than the client logical interface (IP demux or PPPoE).

Reparenting is not supported for enhanced subscriber management logical interfaces in a CoS hierarchical scheduler that includes enhanced subscriber management logical interfaces over a purely dynamic column and enhanced subscriber management logical interfaces over a partially static column. The following describes real-world network environments where reparenting might be required and the alternative approaches that can be adopted in such conditions:

Adding or removing static CoS configuration from an IFL set or an underlying IFL with enhanced subscriber management logical interface on top of it—In such a scenario, adding or removing static CoS is not supported after a subscriber has logged in to the interface column in an environment where

enhanced subscriber management is enabled. A commit error occurs when you attempt this CoS configuration change. This commit failure is not a problem in customer networks because the networks are previously designed, Layer 2 nodes specified, and CoS is configured much before clients are logged in.

Two dynamic profiles for Client logical interfaces over a single CVLAN (or an ACI VLAN) with underlying CoS configuration in one client profile and not in the other profile—In such a scenario, you can maintain dynamic profiles with underlying configuration to be consistent – either all profiles contain underlying CoS config or none of them contain CoS configuration. A negative acknowledgment is sent when a subscriber attempts to log in if a differing way of CoS configuration is observed in the client profiles.

A client profile for an internal node (for example, C-VLAN or IFL set) that does not contain CoS initially and CoS is applied later using a service profile—In such a scenario, it is required that you always specify CoS scheduling in the client profile if you want to reapply some of the settings using a service profile. If this method of configuration is not adopted, a negative acknowledgment is sent when a subscriber attempts to log in. Static or dynamic demux, PPPoE, or PPP interfaces over aggregated Ethernet logical interfaces are not supported.

Consider a scenario in which three subscriber queues, namely, PPPoE subscriber queue 1, PPPoE subscriber queue 2, and DHCP subscriber queues, are established. A Gigabit Ethernet interface, ge-1/0/0 is at Layer 1. Two Layer 2 interface nodes are stacked over the Layer 1 base interface. The Layer 2 interfaces are ge-1/0/0.x or demux0.x and ge-1/0/0.y or demux0.y. Logical interface sets, pppoe-iflset (for access node) and demux-iflset (for home network), are configured at Layer 3 to handle two sets of PPPoE subscriber queues respectively over the Layer 2 interface, ge-1/0/0.x or demux0.x. A traffic control profile, subscriber-tcp, is attached to both these Layer 3 IFL sets. ppp-demux-iflset (demux and pppoe) is the interface set over the Layer 2 interface of ge-1/0/0.y or demux0.y. A traffic control profile, subscriber-tcp, is attached to this interface set. ge-1/0/0.X or demux0.X is the UIFL for all logical interfaces that belong to the pppoe-iflset and demux-iflset. In this topology, ge-1/0/0.Y or demux0.Y is the UIFL for all logical interface that belong to ppp-demux-iflset.

Consider another scenario in which three susbcriber queues, PPPoE subscriber queues, demux subscriber queues, and DHCP subscriber queues, are established. A Gigabit Ethernet interface, ge-1/0/0 is at Layer 1. Two Layer 2 interface nodes are stacked over the Layer 1 base interface. The Layer 2 interfaces are ge-1/0/0.X or demux0.X, and ge-1/0/0.Y or demux0.Y. At Layer 3, pp0.XX is configured over the underlying Layer 2 interface of ge-1/0/0.X or demux0.X, demux0.ZZ is configured over the underlying Layer 2 interface of ge-1/0/0.X or demux0.X, and pp0.YY is configured over the underlying Layer 2 interface of ge-1/0/0.Y or demux0.Yge- 1/0/0.Y or demux0.Y. Traffic control profiles, subcriber-tcp, are applied to pp0.xx for PPPoE subscriber queues, to demux0.yy for demux subscriber queues, and pp0.yy for DHCP subscriber queues. In this topology, ge-1/0/0.X or demux0.X is the underlying IFL for pp0.XX and demux0.ZZ. ge-1/0/0.Y or demux0.Y is the underlying IFL for pp0.YY.

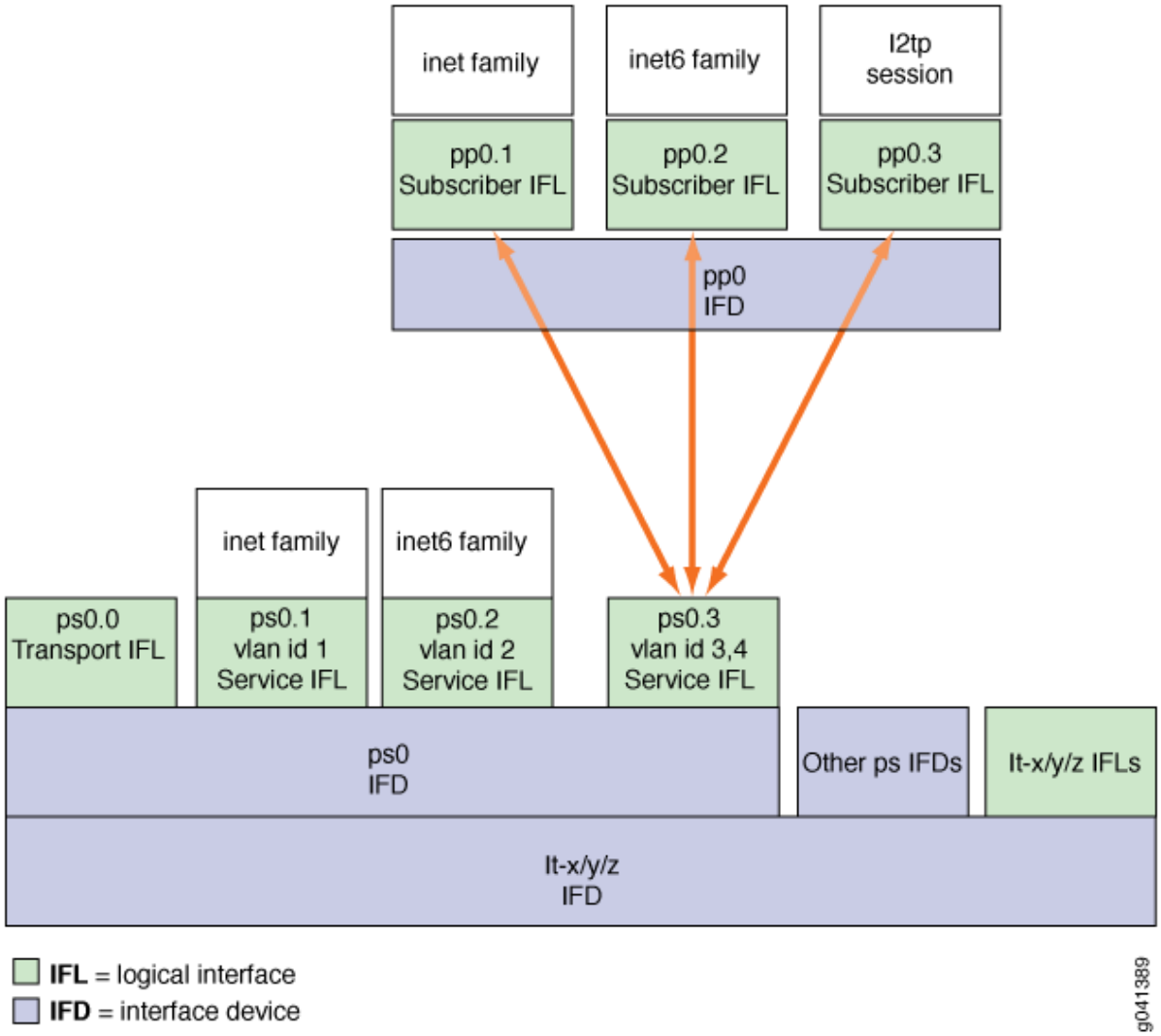
Enhanced Subscriber Management Subscriber Logical Interfaces or Interface Sets Over MPLS Pseudowires for a CoS scheduler Hierarchy

Starting in Junos OS Release 15.1, you can enable a CoS scheduling hierarchy for subscriber logical interfaces or interface sets over underlying MPLS pseudowire logical interfaces. Until Junos OS Release 14.2, an interface set can be either at Layer 2 or Layer 3 levels of the CoS three-level hierarchical scheduler. When the interface set is at the Layer 3 level, a mechanism to configure the Layer 2 node to which the Layer 3 node belonged was not available. As a result, the Layer 2 node was a dummy node in such a case for the three-level hierarchical scheduler.

In certain Broadband Remote Access Server (B-RAS) deployments, when you use an interface set to denote a home network, it might be necessary to configure the home network and the access node (such as a digital subscriber line access multiplexer, or DSLAM) in a scheduler hierarchy. This method of hierarchical scheduler is necessary in agent circuit identifier (ACI) VLANs because a home or an ACI is always an interface set in such topologies.

Enhanced subscriber management enables you to take advantage of increased scaling and performance for configuring and managing dynamic interfaces and services for subscriber management. You can now enable an enhanced subscriber management logical interface, such as an MPLS pseudowire logical interface to function as a Layer 2 node in a CoS hierarchical scheduler. A subscriber logical interface is considered to operate at Layer 2 only if you configure three-level hierarchical scheduling on the logical tunnel anchor point on the physical interface (the IFD). An MPLS pseudowire is a virtual device that is stacked above the logical tunnel anchor point. Implicit hierarchy processes the interface stack properly in such a setup. To configure three-level hierarchical scheduling, include the `implicit- hierarchy` option at the `[edit interfaces “$junos-interface-ifd-name” hierarchical-scheduler]` or the `[edit interfaces lt-device hierarchical-scheduler]` hierarchy level. If the `implicit-hierarchy` option is not set on the logical tunnel anchor point, logical interfaces behave normally with the hierarchical-scheduler mode configured with or without the `hierarchical-scheduler maximum-hierarchy-levels` option under the `[edit interfaces` *`interface-name`* `hierarchical-scheduler]` statement. Figure 1 on page 93 shows the protocol stack for a pseudowire subscriber logical interface.

Figure 1: Pseudowire Subscriber Interface Protocol Stack



In this case, when you apply a traffic-control profile to the pseudowire and service logical interfaces, they both reside in level 3 scheduler nodes and do not form a scheduling hierarchy, which might not be the desirable behavior. Subscriber logical interfaces at Layer 3 that are stacked over the underlying MPLS pseudowire logical interfaces at Layer 2 are supported if the Layer 2 logical interface is an underlying interface of the Layer 3 interface.

For example, if a PPPoE logical interface contains an MPLS pseudowire, psps-anchor-device-name.logical-unit-number, as the underlying interface, the psps-anchor-device-name.logical-unit-number interface can be at Layer 2 and the PPPoE logical interface can be at Layer 3. You can also configure PPP or IP demux interfaces in such a fashion at Layer 3. Similarly, you can configure MPLS pseudowire logical interfaces at Layer 2 that serve as underlying interfaces for logical interface sets, such as PPPoE ACI interface sets or IP demux interface sets, where all the member logical interfaces of

the interface set contain the same underlying MPLS pseudowire at Layer 2. You can configure the MPLS pseudowire logical interfaces at Layer 2 in a dynamic profile or in a static CoS configuration.

Dynamic profile CoS configuration for underlying logical interfaces is supported because two interface stanzas with TCPs in one dynamic profile are considered valid. For dynamic pseudowire underlying logical interfaces, you can configure in a profile different from the client logical interface profile or in the same profile as the client profile. If the underlying logical interface is static and CoS is configured dynamically in a dynamic profile, it must be specified in the same profile as the client logical interface. However, CoS for the underlying logical interfaces must be configured either in a dynamic profile or in a static CoS because both static CoS and dynamic CoS are not supported on the same logical interface.

Reparenting is a technique that denotes the movement of the CoS hierarchical scheduler from one node to another node, such as moving all logical interfaces stacked over an underlying logical interface on top of the base physical interface to be over the underlying logical interface directly and adding the scheduling node. This movement might occur when when CoS for the underlying logical interface or the underlying interface set is configured later than the client logical interface (IP demux or PPPoE).

Reparenting is not supported for enhanced subscriber management logical interfaces in a CoS hierarchical scheduler that includes enhanced subscriber management logical interfaces over a purely dynamic column and enhanced subscriber management logical interfaces over a partially static column. The following describes real-world network environments where reparenting might be required and the alternative approaches that can be adopted in such conditions:

Adding or removing static CoS configuration from a logical interface (IFL) set or an underlying IFL with enhanced subscriber management logical interface on top of it is not supported. In such a scenario, adding or removing static CoS is not supported after a subscriber has logged in to the interface column in an environment where enhanced subscriber management is enabled. A commit error occurs when you attempt this CoS configuration change. This commit failure is not a problem in customer networks because the networks are previously designed, Layer 2 nodes specified, and CoS is configured much before clients are logged in.

Two dynamic profiles for Client logical interfaces over a single CVLAN (or an ACI VLAN) with underlying CoS configuration in one client profile and not in the other profile—In such a scenario, you can maintain dynamic profiles with underlying configuration to be consistent – either all profiles contain underlying CoS config or none of them contain CoS configuration. A negative acknowledgment is sent when a subscriber attempts to log in if a differing way of CoS configuration is observed in the client profiles.

A client profile for an internal node (for example, C-VLAN or IFL set) that does not contain CoS initially and CoS is applied later using a service profile—In such a scenario, it is required that you always specify CoS scheduling in the client profile if you want to reapply some of the settings using a service profile. If this method of configuration is not adopted, a negative acknowledgment is sent when a subscriber attempts to log in. Static or dynamic demux, PPPoE, or PPP interfaces over aggregated Ethernet logical interfaces are not supported.

Consider a scenario in which three subscriber queues, namely, PPPoE subscriber queue 1, PPPoE subscriber queue 2, and DHCP subscriber queues, are established. A logical tunnel interface, lt-1/0/0 is

at Layer 1. Two Layer 2 interface nodes are stacked over the Layer 1 base interface. The Layer 2 interfaces are psX.Y and psX.Z. Logical interface sets, ppp0.XX (for access node) and demux0.ZZ (for home network), are configured at Layer 3 to handle PPPoE subscriber queues and DHCP subscriber queues respectively over the Layer 2 interface, psX.Y. A logical interface, pp0.YY, is configured at Layer 3 to handle PPPoE subscriber queues over the Layer 2 interface, psX.Z. A traffic control profile, subscriber-tcp, is attached to these Layer 3 interfaces. psX.Y is the underlying logical interface for pp0.XX and demux0.ZZ if Y is not 0. psX.Z is the underlying logical interface for pp0.YY if Z is not 0. psX.0 is called the pseudowire transport logical interface and psX.Y (where Y is not equal to 0) is called the pseudowire service logical interface.

Consider another scenario in which two susbcriber queues, PPPoE subscriber queues and DHCP subscriber queues, are established. A logical tunnel interface, lt- 1/0/0 is at Layer 1. Two Layer 2 interface nodes are stacked over the Layer 1 base interface. The Layer 2 interfaces are psX.Y and psX.Z. Logical interface sets, pppoe-iflset (for access node) and demux-iflset (for home network), are configured at Layer 3 to handle PPPoE subscriber queues and DHCP subscriber queues respectively over the Layer 2 interface, psX.Y. A logical interface set, ppp-demux-iflset, is configured at Layer 3 to handle PPPoE and DHCP subscriber queues over the Layer 2 interface, psX.Z. A traffic control profile, subscriber-tcp, is attached to these Layer 3 interfaces. psX.Y is the underlying logical interface for all logical interfaces that belong to the pppoe-iflset and demux-iflset if Y is not equal to 0. psX.Z is the underlying logical interface for all logical interfaces that belong to the ppp-demux-iflset interface set if Z is not 0. psX.0 is called the pseudowire transport logical interface and psX.Y (where Y is not equal to 0) is called the pseudowire service logical interface.

Configuring Layer 2 Subscriber Logical Interfaces for CoS Hierarchical Schedulers Using Dynamic Profiles for Differentiating Home and Access Node Networks

In certain Broadband Remote Access Server (B-RAS) deployments, when you use an interface set to denote a home network, it might be necessary to configure the home network and the access node (such as a digital subscriber line access multiplexer, or DSLAM) in a scheduler hierarchy. This method of hierarchical scheduler is necessary in agent circuit identifier (ACI) VLANs because a home or an ACI is always an interface set in such topologies. You can configure a subscriber logical interface or an interface set at Layer 3 over an underlying enhanced subscriber management logical interface that functions as a Layer 2 node. You can configure a the Layer 2 logical interface in a CoS dynamic profile.

Before you apply CoS attributes to VLANs:

- Create a basic dynamic profile.

  See *Configuring a Basic Dynamic Profile*.

Consider a scenario in which a Layer 3 interface set, ACI-set aci-1006-ps0.3221225479, is stacked over dynamic a MPLS pseudowire service logical interface, ps0.3221225479, at Layer 2. You can configure only one traffic-control-profile under a dynamic profile. You must define the output-traffic-control-profile that binds the traffic-control profile to the interface within the same dynamic profile as the interface. Two traffic control profiles are defined to apply an output traffic scheduling and shaping profile to the MPLS pseudowire logical interface. These control profiles are an-tcp to be applied for TCP subscribers that are terminated at the access mode and an-tcp-remaining, which is a remaining traffic-control profile to a logical interface to provide minimal CoS scheduling when you have not configured or over-provisioned Layer 3 schedulers.

To apply CoS attributes, such as shaping, at the household level, you must set and define the CoS policy for the agent-circuit-identifier VLAN interface set using the dynamic profile for the agent-circuit-identifier interface set (not the subscriber profile). You can also configure a traffic-control profile and a remaining traffic-control profile for a dynamic interface set.

The following example is a CoS profile for an ACI set using a unique-ID based dynamic scheduler map:

Configure a CoS dynamic profile with a simple traffic-control profile that is applied to the dynamic interface set that represents the ACI VLAN.

1. Configure CoS to support a dynamic interface set in the CoS profile:

```
[edit dynamic-profiles profile-name]
user@host# edit interface "$junos-interface-name"
```

2. Configure the interfaces.

```
[edit dynamic-profiles profile-name interfaces]
user@host# edit interface-set "$junos-interface-set-name"
user@host# edit interface "$junos-interface-ifd-name"
```

3. Configure the CoS traffic-control profile.

```
[edit class-of-service]
user@host# edit traffic-control-profiles traffic-control-profile-name
user@host# set shaping-rate rate
user@host# set guaranteed-rate rate
```

4. Specify the output traffic control profile and the remaining traffic control profile for the underlying logical interfaces that are members of the interface set.

```
[edit class-of-service interfaces]
user@host# edit interface "$junos-interface-ifd-name" unit "$junos-underlying-interface-unit"
user@host# edit output-traffic-control-profile profile-name
user@host# edit output-traffic-control-profile-remaining profile-name
```

5. Specify the output traffic control profile for the interface set.

```
[edit class-of-service interfaces]
user@host# edit interface-set "$junos-interface-set-name"
user@host# edit output-traffic-control-profile profile-name
```

The following example is a CoS profile for an ACI set using a unique ID-based dynamic scheduler map:

```
aci-set-profile {
    variables {
        ds1q0q2DP uid;
        ds1q1q2DP uid;
        be1_dp uid;
        ef1_dp uid;
        af1_dp uid;
        nc1_dp uid;
    }
    interfaces {
        interface-set "$junos-interface-set-name" {
            interface "$junos-interface-ifd-name";
        }
    }
    class-of-service {
        traffic-control-profiles {
            tcp2 {
                inactive: scheduler-map ss1q0q1DP;
                shaping-rate 50m;
                guaranteed-rate 30m;
                overhead-accounting bytes -20;
            }
            tcp3 {
                scheduler-map "$ds1q1q2DP";
                shaping-rate 30m;
```

```
            guaranteed-rate 10m;
            overhead-accounting bytes -20;
        }
    }
    interfaces {
        interface-set "$junos-interface-set-name" {
            output-traffic-control-profile tcp2;
            output-traffic-control-profile-remaining tcp3;
        }
    }
    scheduler-maps {
        "$ds1q0q2DP" {
            forwarding-class be scheduler "$be1_dp";
            forwarding-class af scheduler "$af1_dp";
            forwarding-class nc scheduler "$nc1_dp";
        }
        "$ds1q1q2DP" {
            forwarding-class ef scheduler "$ef1_dp";
            forwarding-class af scheduler "$af1_dp";
            forwarding-class nc scheduler "$nc1_dp";
        }
    }
    schedulers {
        "$be1_dp" {
            transmit-rate percent 25;
            priority low;
            drop-profile-map loss-priority low protocol any drop-profile d3;
            drop-profile-map loss-priority medium-low protocol any drop-profile d2;
            drop-profile-map loss-priority medium-high protocol any drop-profile d1;
            drop-profile-map loss-priority high protocol any drop-profile d0;
        }
        "$ef1_dp" {
            transmit-rate percent 25;
            priority low;
            drop-profile-map loss-priority low protocol any drop-profile d3;
            drop-profile-map loss-priority medium-low protocol any drop-profile d2;
            drop-profile-map loss-priority medium-high protocol any drop-profile d1;
            drop-profile-map loss-priority high protocol any drop-profile d0;
        }
        "$af1_dp" {
            transmit-rate percent 25;
            priority low;
            drop-profile-map loss-priority low protocol any drop-profile d3;
```

```
                drop-profile-map loss-priority medium-low protocol any drop-profile d2;
                drop-profile-map loss-priority medium-high protocol any drop-profile d1;
                drop-profile-map loss-priority high protocol any drop-profile d0;
            }
            "$nc1_dp" {
                transmit-rate percent 25;
                priority low;
                drop-profile-map loss-priority low protocol any drop-profile d3;
                drop-profile-map loss-priority medium-low protocol any drop-profile d2;
                drop-profile-map loss-priority medium-high protocol any drop-profile d1;
                drop-profile-map loss-priority high protocol any drop-profile d0;
            }
        }
    }
}
```

You can use the `show class-of-service scheduler-hierarchy interface` *`interface-name`* command to verify the CoS hierarchical schedulers configured on the interfaces. For example, the following output illustrates that ACI-set aci-1003-demux0.3221225482 is stacked over demux0.3221225482.

```
user@host> show class-of-service scheduler-hierarchy interface ge-0/2/0
Interface/                    Shaping Guarnteed  Guaranteed/   Queue   Excess
Resource name                    rate      rate       Excess  weight   weight
                                kbits     kbits     priority          high/low
ge-0/2/0                      1000000
  ge-0/2/0 RTP                1000000         0                          1    1
    best-effort               1000000         0     Low  Low     950
    network-control           1000000         0     Low  Low      50
  demux0.3221225482            100000     80000                        500  500
    demux0.3221225482 RTP
                                30000     20000                        125  125
      best-effort               30000     19000     Low  Low     950
      network-control           30000      1000     Low  Low      50
    aci-1003-demux0.3221225482     out-of-scheduler-resources
```

From the following sample output, you can verify that ACI-iflset aci-1001-ps1.3221225472 is stacked over a static pseudowire transport logical interface, ps1.0

```
user@host> show class-of-service scheduler-hierarchy interface ps1
Interface/                    Shaping Guarnteed  Guaranteed/   Queue   Excess
```

```
Resource name                   rate      rate       Excess  weight   weight
                               kbits     kbits     priority          high/low
lt-0/3/0                    10000000
  lt-0/3/0 RTP              10000000         0                          1    1
    best-effort             10000000         0     Low  Low     950
    network-control         10000000         0     Low  Low      50
  ps1.0                       100000         0                          1    1
    ps1.0 RTP                 500000         0                          1    1
      best-effort             400000         0     Low  Low    1000
    aci-1001-ps1.3221225472
                              200000     10000                        500  500
      best-effort             160000      2000     Low  Low    1000
```

From the following sample output, you can verify that ACI-set aci-1006-ps0.3221225479 is stacked over the dynamic pseudowire service logical interface, ps0.3221225479.

```
user@host> show class-of-service scheduler-hierarchy interface ps0
Interface/                    Shaping Guarnteed  Guaranteed/   Queue   Excess
Resource name                    rate      rate       Excess  weight   weight
                                kbits     kbits     priority          high/low
lt-0/3/0                    10000000
  lt-0/3/0 RTP              10000000         0                          1    1
    best-effort             10000000         0     Low  Low     950
    network-control         10000000         0     Low  Low      50
  ps0.32767                 10000000      2000                         50   50
    best-effort             10000000      1900     Low  Low     950
    network-control         10000000       100     Low  Low      50
  ps0.3221225479              100000         0                          1    1
    ps0.3221225479 RTP         40000     20000                        500  500
      best-effort               5000      3000  Medium  Low       1
      expedited-forwarding     40000      2000  Medium High    1000
    aci-1006-ps0.3221225479
                              100000     10000                        250  250
      best-effort               5000      1500  Medium  Low       1
      expedited-forwarding    100000      1000  Medium High     500
      assured-forwarding      100000      1000  Medium High     500
      network-control         100000      2000    High High       1
```

RELATED DOCUMENTATION

Guidelines for Configuring Dynamic CoS for Subscriber Access | 44

Changing CoS Services Overview | 166

Example: Configuring Layer 2 Subscriber Logical Interfaces for CoS Hierarchical Schedulers Using Static CoS for Differentiating Home and Access Node Networks

IN THIS SECTION

- Requirements | 101
- Overview | 102
- Configuration | 102
- Verification | 105

Starting in Junos OS Release 15.1, in certain Broadband Remote Access Server (B-RAS) deployments, when you use an interface set to denote a home network, it might be necessary to configure the home network and the access node (such as a digital subscriber line access multiplexer, or DSLAM) in a scheduler hierarchy. This method of hierarchical scheduler is necessary in agent circuit identifier (ACI) VLANs because a home or an ACI is always an interface set in such topologies. You can enable an enhanced subscriber management logical interface, such as an MPLS pseudowire logical interface to function as a Layer 2 node in a CoS hierarchical scheduler. A subscriber logical interface is considered to operate at Layer 2 only if you configure CoS three-level hierarchical scheduling on the logical tunnel anchor point on the physical interface (the IFD). When you include the implicit-hierarchy option, a hierarchical relationship is formed between the CoS scheduler nodes at level 1, level 2, and level 3. The implicit-hierarchy option is supported only on MPC/MIC subscriber interfaces and interface sets running over aggregated Ethernet on MX Series routers.

Requirements

This example uses the following hardware and software components:

- Junos OS Release 15.1
- MX Series Router with MPCs

Overview

You specify an anchor point, which identifies the logical tunnel interface that terminates the pseudowire tunnel at the access node. Consider a scenario in which lt-0/3/0 is the logical tunnel interface, and an MPLS pseudowire transport logical interface, ps1.0, that is anchored on the logical tunnel. Three-level hierarchical scheduling is enabled on the logical tunnel interface for static CoS configuration.

Configuration

IN THIS SECTION

- CLI Quick Configuration | **102**
- Configuring an MPLS Pseudowire Transport Logical Interface Over a Logical Tunnel in a Static CoS Setup | **103**
- Results | **104**

To configure an enhanced subscriber management logical interface, such as an MPLS pseudowire logical interface to function as a Layer 2 node in a CoS hierarchical scheduler. perform these tasks:

CLI Quick Configuration

To quickly configure the MPLS pseudowire logical interface to function as a Layer 2 node in a three-level hierarchical scheduler, copy the following commands and paste them into the router terminal window:

```
[edit]
set interfaces lt-0/3/0
set interfaces lt-0/3/0 hierarchical-scheduler implicit-hierarchy
set interfaces ps1
set  interfaces ps1 description client-port-l2circuit
set  interfaces ps1 flexible-vlan-tagging
set  interfaces ps1 anchor-point lt-0/3/0
set  interfaces ps1 unit 0
set  interfaces ps1 unit 0 encapsulation ethernet-ccc
set  interfaces ps1 unit 0 output-traffic-control-profile profile-name
```

Configuring an MPLS Pseudowire Transport Logical Interface Over a Logical Tunnel in a Static CoS Setup

Step-by-Step Procedure

Three-level scheduling on pseudowire logical interfaces over a transport logical interface requires you to apply the traffic-control profiles at both the pseudowire logical interface and the pseudowire transport logical interface. To configure three-level scheduling on pseudowire transport logical interfaces over a logical tunnel physical interface (LT ifd):

1. Configure the hierarchical scheduler for the physical interface used for the logical tunnel (anchor point). For three-level scheduling the hierarchical scheduler must be set to `implicit-hierarchy`.

```
[edit]
user@host#edit interfaces lt-0/3/0
user@host#set hierarchical-scheduler implicit-hierarchy
```

2. Specify that you want to configure the pseudowire subscriber logical interface device.

```
 [edit]
user@host# edit interfaces ps1
```

3. Configure a description for the pseudowire subscriber logical interface.

```
[edit interfaces ps1]
user@host# set  description client-port-l2circuit
```

4. Specify the `flexible-vlan-tagging` statement to indicate that this interface is for use with both VLAN and stacked VLAN ranges.

```
[edit interfaces ps1]
user@host# set  flexible-vlan-tagging
```

5. Specify the logical tunnel (lt) interface that identifies the Packet Forwarding Engine that processes the pseudowire termination.

```
[edit interfaces ps1]
user@host# set anchor-point lt-0/3/0
```

6. Specify that you want to configure unit 0, which represents the transport logical interface.

```
[edit interfaces ps1]
user@host# edit unit 0
```

7. Specify the ethernet-ccc encapsulation method for the transport logical interface.

```
[edit interfaces ps0 unit 0]
user@host# set encapsulation ethernet-ccc
```

8. Specify the traffic-control profile to use on the pseudowire transport logical interface.

```
[edit class-of-service]
user@host#edit interfaces ps 1
user@host#edit unit 0
user@host#set output-traffic-control-profile profile-name
```

Results

In configuration mode, confirm your configuration by entering the `show` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
lt-0/3/0 {
    hierarchical-scheduler implicit-hierarchy;
}

ps1 {
    description client-port-l2circuit;
    anchor-point {
        lt-0/3/0;
    }
    flexible-vlan-tagging;
    unit 0 {
        encapsulation ethernet-ccc;
    }
}
```

Verification

IN THIS SECTION

- Verifying the Scheduler Hierarchy Configured on the Interfaces | **105**

Confirm that the configuration is working properly.

Verifying the Scheduler Hierarchy Configured on the Interfaces

Purpose

Verify the CoS hierarchical scheduler configured on the Layer 2 and Layer 3 interface nodes.

Action

From operational mode, enter the `show class-of-service scheduler-hierarchy interface ps0` command.

```
user@host> show class-of-service scheduler-hierarchy interface ps0

Interface/                    Shaping Guarnteed  Guaranteed/   Queue   Excess
Resource name                    rate      rate       Excess  weight   weight
                                kbits     kbits     priority          high/low
lt-0/3/0                     10000000
  lt-0/3/0 RTP               10000000         0                          1    1
    best-effort              10000000         0     Low  Low     950
    network-control          10000000         0     Low  Low      50
  ps0.0                        200000         0                          1    1
    ps0.0 RTP                10000000         0                          1    1
      best-effort            10000000         0     Low  Low     950
      network-control        10000000         0     Low  Low      50
    ps0.3221225474             100000         0                          1    1
      best-effort                5000         0  Medium  Low    1000



user@host> show class-of-service scheduler-hierarchy interface ps0
```

```
Interface/                    Shaping Guarnteed  Guaranteed/   Queue   Excess
Resource name                    rate      rate       Excess  weight   weight
                                kbits     kbits     priority          high/low
lt-0/3/0                     10000000
  lt-0/3/0 RTP               10000000         0                         1    1
    best-effort              10000000         0     Low  Low     950
    network-control          10000000         0     Low  Low      50
  ps0.32767                  10000000      2000                        33   33
    best-effort              10000000      1900     Low  Low     950
    network-control          10000000       100     Low  Low      50
  ps0.3221225474               200000         0                         1    1
    ps0.3221225474 RTP         100000     30000                       500  500
      best-effort               30000      3000  Medium  Low     250
      expedited-forwarding      32000      9000     Low  Low     750
    pp0.3221225475             100000     10000                       166  166
      best-effort                5000      1500     Low  Low       1
      expedited-forwarding     100000      1000  Medium High     500
      assured-forwarding       100000      1000  Medium High     500
      network-control          100000      2000    High High       1
```

Meaning

Shows that dynamic pseudowire service logical interface, ps0.3221225474, is stacked over the static pseudowire transport logical interface, ps0.0. Also, the sample output denotes that pp0.3221225475 is stacked over dynamic pseudowire service logical interface, ps0.3221225474.

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, in certain Broadband Remote Access Server (B-RAS) deployments, when you use an interface set to denote a home network, it might be necessary to configure the home network and the access node (such as a digital subscriber line access multiplexer, or DSLAM) in a scheduler hierarchy.

CHAPTER 6

Configuring Dedicated Queue Scaling with Hierarchical CoS or Per-Unit Scheduling

IN THIS CHAPTER

- Dedicated Queue Scaling for CoS Configurations on MIC and MPC Interfaces Overview | **107**
- Managing Dedicated and Remaining Queues for Dynamic CoS Configurations on MIC and MPC Interfaces | **110**
- Verifying the Number of Dedicated Queues Configured on MIC and MPC Interfaces | **113**

Dedicated Queue Scaling for CoS Configurations on MIC and MPC Interfaces Overview

IN THIS SECTION

- Queue Scaling for MPCs | **108**
- Managing Remaining Queues | **109**

Queuing Ethernet Modular Port Concentrators (MPCs) provide a set of dedicated queues for subscriber interfaces configured with hierarchical scheduling or per-unit scheduling.

The dedicated queues offered on these MPCs enable service providers to reduce costs through different scaling configurations. These queuing MPCs enable service providers to reduce the cost per subscriber by allowing many subscriber interfaces to be created with four or eight queues.

This topic describes the overall queue, scheduler node, and *logical interface* scaling for subscriber interfaces created on these MIC and MPC combinations.

Queue Scaling for MPCs

Beginning with Junos OS Release 15.1, MPC2E-3D-NG-Q, MPC3E-3D-NG-Q, MPC5EQ-40G10G, and MPC5EQ-100G10G MPCs support up to five levels of hierarchical queuing. Beginning with Junos OS Release 16.1R1, MPC7 line cards also support five levels of hierarchical queuing. Table 13 on page 108 lists the number of dedicated queues and nodes supported per MPC.

Table 13: Dedicated Queues for MPCs

MPC	Dedicated Queues	Level 4 Nodes	Level 3 Nodes	Level 2 Nodes	Level 1 Nodes (Ports)
MPC2E-3D-NG-Q MPC3E-3D-NG-Q	512,000	64,000	16,000	4000	384
MPC5EQ-40G10G MPC5EQ-100G10G	1 million	128,000	32,000	4000	384
MPC7	512,000	64,000	16000	8000	252
MPC10E on MX10K series platforms	256,000	32,000	8,000	4,000	128

CAUTION: The maximum scaling targets provided in Table 13 on page 108 are based on system level design specifications. Actual realized subscriber or session scale is highly dependent upon the configuration and can be influenced by configuration variables including: the number of routes, the number of enabled services, the number of policy and firewall filters, policers, counters, statistics and access model type. Once you define a configuration, your Juniper account team can help characterize the expected system level scale or scale range for your live deployment.

MPCs vary in the number of Packet Forwarding Engines on board. MPC2E-3D-NG-Q and MPC3E-3D-NG-Q MPCs each have one Packet Forwarding Engine, allowing all 64,000 level 4 (subscriber) nodes to be allocated to a single MIC. MPC5EQ MPCs have two Packet Forwarding Engines, one for each

possible MIC, each supporting 64,000 level 4 (subscriber) nodes. MPC7 MPCs also have two Packet Forwarding Engines, one for each possible MIC, each supporting 256,000 dedicated queues and 32,000 level 4 (subscriber) nodes.

NOTE: The nonqueuing MPCs MPC2E-3D-NG, MPC3E-3D-NG, MPC5E-40G10G, and MPC5E-100G10G provide up to eight queues per port in standard configuration. However, each of these MPCs can be configured to provide limited-scale hierarchical class of service (HCoS) and up to 32,000 queues.

Managing Remaining Queues

In Junos OS releases earlier than Release 15.1R4, SNMP traps generate system log messages to notify you:

- When the number of available dedicated queues on the MPC drops below 10 percent. For example:

```
Mar 15 14:55:22.977 host cosd[1963]: COSD_OUT_OF_DEDICATED_QUEUES: Queue usage count for
interface xe-3/0/0 is at 90 percent
```

- When the maximum number of dedicated queues on the MPCs is reached. For example,

```
Mar 15 18:01:59.344 host cosd[3848]: COSD_OUT_OF_DEDICATED_QUEUES: Queue usage count for
interface xe-3/0/0 is at 100 percent.
```

  When the maximum number of dedicated queues is allocated, the system does not provide subsequent subscriber interfaces with a dedicated set of queues. For per-unit scheduling configurations, there are no configurable queues remaining on the MPC.

For hierarchical scheduling configurations, remaining queues are available when the maximum number of dedicated queues is reached on the MPC. Traffic from these logical interfaces is considered unclassified and attached to a common set of queues that are shared by all subsequent logical interfaces. These common queues are the default port queues that are created for every port. You can configure a traffic-control profile and attach that to the interface to provide CoS parameters for the remaining queues. These subscriber interfaces remain with this traffic-control profile, even if dedicated queues become available.

NOTE: Starting in Junos OS Release 15.1R4, the COSD_OUT_OF_DEDICATED_QUEUES functionality is not available for QoS-enabled dynamic subscribers. Starting in Junos OS Release 17.4R1, CoS resource monitoring

enables you to set a per-FPC queue threshold of up to 90 percent of resources bound to a scheduling hierarchy; subscriber logins are not allowed when the threshold is reached. However, this threshold applies to all queues, not dedicated queues alone. See Resource Monitoring for Subscriber Management and Services Overview for more information.

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
16.1R1	Beginning with Junos OS Release 16.1R1, MPC7 line cards also support five levels of hierarchical queuing.
15.1R1	Beginning with Junos OS Release 15.1, MPC2E-3D-NG-Q, MPC3E-3D-NG-Q, MPC5EQ-40G10G, and MPC5EQ-100G10G MPCs support up to five levels of hierarchical queuing.

RELATED DOCUMENTATION

- Hierarchical Class of Service User Guide
- Understanding Hierarchical Scheduling
- Managing Dedicated and Remaining Queues for Static CoS Configurations on MIC and MPC Interfaces
- Understanding Hierarchical Scheduling for MIC and MPC Interfaces

Managing Dedicated and Remaining Queues for Dynamic CoS Configurations on MIC and MPC Interfaces

IN THIS SECTION

- Configuring the Maximum Number of Queues for MIC and MPC Interfaces | **111**
- Configuring Remaining Common Queues on MIC and MPC Interfaces | **111**

This topic describes how to manage dedicated and remaining queues for static and dynamic subscriber interfaces configured in dynamic profiles.

You manage queues at the chassis and physical port level in the static configuration hierarchies, then configure dynamic scheduling and shaping parameters for the subscriber interfaces in the dynamic profile.

Configuring the Maximum Number of Queues for MIC and MPC Interfaces

30-Gigabit Ethernet Queuing MPCs and 60-Gigabit Ethernet Queuing and Enhanced Queuing MPCs support a dedicated number of queues when configured for hierarchical scheduling and per-unit scheduling configurations.

To scale the number of subscriber interfaces per queue, you can modify the number of queues supported on the MIC.

To configure the number of queues:

1. Specify that you want to configure the MIC.

```
user@host# edit chassis fpc slot-number pic pic-number
```

2. Configure the number of queues.

```
[edit chassis fpc slot-number pic pic-number]
user@host# set max-queues-per-interface (8 | 4)
```

SEE ALSO

Dedicated Queue Scaling for CoS Configurations on MIC and MPC Interfaces Overview | 107

Configuring Remaining Common Queues on MIC and MPC Interfaces

30-Gigabit Ethernet Queuing MPCs and 60-Gigabit Ethernet Queuing and Enhanced Queuing MPCs support a dedicated set of queues when configured with hierarchical scheduling.

When the number of dedicated queues is reached on the module, there can be queues remaining. Traffic from these logical interfaces are considered unclassified and attached to a common set of queues that are shared by all subsequent logical interfaces.

You can configure traffic shaping and scheduling resources for the remaining queues by attaching a special traffic-control profile to the interface. This feature enables you to provide the same shaping and scheduling to remaining queues as the dedicated queues.

To configure the remaining queues on a MIC or MPC interface:

1. Configure CoS parameters in a traffic-control profile.

```
[edit class-of-service]
user@host# edit traffic-control-profiles profile-name
```

2. Enable hierarchical scheduling for the interface.

```
[edit interfaces interface-name]
user@host# set hierarchical-scheduler
```

3. Attach the traffic control profiles for the dedicated and remaining queues to the port on which you enabled hierarchical scheduling.

   To provide the same shaping and scheduling parameters to dedicated and remaining queues, reference the same traffic-control profile.

   a. Attach the traffic-control profile for the dedicated queues on the interface.

```
[edit class-of-service interfaces interface-name]
user@host# set output-traffic-control-profile profile-name
```

   b. Attach the traffic-control profile for the remaining queues on the interface.

```
[edit class-of-service interfaces interface-name]
user@host# set output-traffic-control-profile-remaining profile-name
```

SEE ALSO

Dedicated Queue Scaling for CoS Configurations on MIC and MPC Interfaces Overview | 107

RELATED DOCUMENTATION

Verifying the Number of Dedicated Queues Configured on MIC and MPC Interfaces | 113

Dedicated Queue Scaling for CoS Configurations on MIC and MPC Interfaces Overview | 107

Configuring Static Hierarchical Scheduling in a Dynamic Profile

Verifying the Number of Dedicated Queues Configured on MIC and MPC Interfaces

IN THIS SECTION

- Purpose | 113
- Action | 113

Purpose

Display the number of dedicated queue resources that are configured for the logical interfaces on a port.

Action

```
user@host#show class-of-service interface ge-1/1/0
Physical interface: ge-1/1/0, Index: 166
Queues supported: 4, Queues in use: 4
Total non-default queues created: 4
  Scheduler map: <default>, Index: 2
  Chassis scheduler map: <default-chassis>, Index: 4

  Logical interface: ge-1/1/0.100, Index: 72, Dedicated Queues: no
    Shaping rate: 32000
    Object              Name                    Type                  Index
    Scheduler-map       <remaining>                                   0
    Classifier          ipprec-compatibility    ip                    13

  Logical interface: ge-1/1/0.101, Index: 73, Dedicated Queues: no
    Shaping rate: 32000
    Object              Name                    Type                  Index
    Scheduler-map       <remaining>                                   0
    Classifier          ipprec-compatibility    ip                    13

  Logical interface: ge-1/1/0.102, Index: 74, Dedicated Queues: yes
    Shaping rate: 32000
```

```
Object                  Name               Type              Index
Traffic-control-profile <control_tc_prof>  Output            45866
```

RELATED DOCUMENTATION

- Managing Dedicated and Remaining Queues for Static CoS Configurations on MIC and MPC Interfaces
- Managing Dedicated and Remaining Queues for Dynamic CoS Configurations on MIC and MPC Interfaces | **110**

CHAPTER 7

Shaping Downstream Traffic Based on Frames or Cells

IN THIS CHAPTER

- Bandwidth Management for Downstream Traffic in Edge Networks Overview | **115**
- Configuring Dynamic Shaping Parameters to Account for Overhead in Downstream Traffic Rates | **118**
- Example: Configuring Dynamic Shaping Parameters to Account for Overhead in Downstream Traffic Rates | **119**
- Configuring Static Shaping Parameters to Account for Overhead in Downstream Traffic Rates | **124**
- Example: Configuring Static Shaping Parameters to Account for Overhead in Downstream Traffic Rates | **125**
- Setting Shaping Rate and Overhead Accounting Based on PPPoE Vendor-Specific Tags | **128**
- Configuring the Shaping Rate and Overhead Accounting Based on PPPoE Vendor-Specific Tags on Dynamic Subscriber Interfaces | **130**
- Reporting the Effective Shaping Rate for Subscribers | **131**
- Verifying the Effective Shaping Rate Reporting Configuration | **132**

Bandwidth Management for Downstream Traffic in Edge Networks Overview

IN THIS SECTION

- Effective Shaping Rate | **116**
- Shaping Modes | **116**
- Byte Adjustments | **117**
- Relationship with Other CoS Features | **117**

In a subscriber access network, traffic with different encapsulations can be passed downstream to other customer premise equipment (CPE) through the MX Series router. Managing the bandwidth of downstream ATM traffic to Ethernet interfaces can be especially difficult because of the different Layer 2 encapsulations.

The downstream network is not necessarily the directly attached network. In typical broadband network gateway (BNG) configurations, the directly attached network is an Ethernet access network, which provides access to either another frame-based network, or a cell-based network.

The *overhead accounting* feature enables you to shape traffic based on whether the downstream network is a frame-based network, like Ethernet, or a cell-based network, like ATM. It assigns a byte adjustment value to account for different encapsulations.

This feature is available on MIC and MPC interfaces.

Effective Shaping Rate

The shaping-rate, also known as peak information rate (PIR), is the maximum rate for a scheduler node or queue.

The true rate of a subscriber at the access-loop/CPE is a function of:

- The shaping-rate in effect for the subscriber's household, in bits per second.
- Whether the subscriber is connected to a frame-based or cell-based network.
- Number of bytes in each frame that are accounted for by the shaper.

NOTE: Chassis egress-shaping-overhead is not included in the effective rate. `Egress-shaping-overhead` accounts for the physical interface overhead (ISO OSI Layer 1). Effective shaping-rate is a Layer 2 (ISO OSI) rate.

Shaping Modes

There are two modes used for adjusting downstream traffic:

- *Frame shaping mode* is useful for adjusting downstream traffic with different encapsulations. Shaping is based on the number of bytes in the frame, without regard to cell encapsulation or padding overhead. Frame is the default shaping mode on the router.
- *Cell shaping mode* is useful for adjusting downstream cell-based traffic. In cell shaping mode, shaping is based on the number of bytes in cells, and accounts for the cell encapsulation and padding overhead.

When you specify cell mode, the resulting traffic stream conforms to the policing rates configured in downstream ATM switches, reducing the number of packet drops in the Ethernet network.

To account for ATM segmentation, the router adjusts all of the rates by 48/53 to account for 5-byte ATM AAL5 encapsulation. In addition, the router accounts for cell padding, and internally adjusts each frame by 8 bytes to account for the ATM trailer.

Byte Adjustments

When the downstream traffic has different byte sizes per encapsulation, it is useful to configure a *byte adjustment* value to adjust the number of bytes per packet to be included in or excluded from the shaping mechanism. This value represents the number of bytes that are encapsulated and decapsulated by the downstream equipment. For example, to properly account for a 4-byte header stripped by the downstream network, set the overhead-accounting bytes to -4. To properly account for a 12-byte header added by the downstream network, set the overhead-accounting bytes to 12.

We recommend that you specify a byte adjustment value that represents the difference between the CPE protocol overhead and B-RAS protocol overhead.

The system rounds up the byte adjustment value to the nearest multiple of 4. For example, a value of 6 is rounded to 8, and a value of –10 is rounded to –8.

You do not need to configure a byte adjustment value to account for the downstream ATM network. However, you can specify the byte value to account for additional encapsulations or decapsulations in the downstream network.

Relationship with Other CoS Features

Enabling the overhead accounting feature affects the resulting shaping rates, guaranteed rate, and excess rate parameters, if they are configured.

The overhead accounting feature also affects the egress shaping overhead feature that you can configure at the chassis level. We recommend that you use the egress shaping-overhead feature to account for the Layer 2 overhead of the outgoing interface, and use the overhead-accounting feature to account for downstream traffic with different encapsulations and cell-based networks.

When both features are configured, the total byte adjustment value is equal to the adjusted value of the overhead-accounting feature plus the value of the egress-shaping-overhead feature. For example, if the configured byte adjustment value is 40, and the router internally adjusts the size of each frame by 8, the adjusted overhead accounting value is 48. That value is added to the egress shaping overhead of 24 for a total byte adjustment value of 72.

RELATED DOCUMENTATION

Configuring Static Shaping Parameters to Account for Overhead in Downstream Traffic Rates | 124

Configuring Dynamic Shaping Parameters to Account for Overhead in Downstream Traffic Rates | 118

Setting Shaping Rate and Overhead Accounting Based on PPPoE Vendor-Specific Tags | 128

Configuring Dynamic Shaping Parameters to Account for Overhead in Downstream Traffic Rates

You can configure the overhead accounting feature to shape downstream traffic based on either frames or cells.

You can also account for the different byte sizes per encapsulation by configuring a byte adjustment value for the shaping mode.

This feature is supported on MPCs on MX Series routers.

To configure the overhead accounting feature in a dynamic profile:

1. Do one of the following to configure the shaping mode:
   - Specify the shaping mode.

     Frame shaping mode is enabled by default.

     ```
     [edit dynamic-profiles profile-name class-of-service traffic-control-profiles profile-name
     user@host#set overhead-accounting (frame-mode | cell-mode)
     ```

   - Configure a variable for the shaping mode.

     ```
     [edit dynamic-profiles profile-name class-of-service traffic-control-profiles profile-name
     user@host#set overhead-accounting $junos-cos-shaping-mode
     ```

2. (Optional) Do one of the following to configure the byte adjustment value:
   - Specify a byte adjustment value.

     ```
     [edit dynamic-profiles profile-name class-of-service traffic-control-profiles profile-name
     user@host#set overhead-accounting bytes byte-value
     ```

- Configure a variable for the byte adjustment.

```
[edit dynamic-profiles profile-name class-of-service traffic-control-profiles profile-name
user@host#set overhead-accounting bytes $junos-cos-byte-adjust
```

BEST PRACTICE: We recommend that you specify a byte adjustment value that represents the difference between the customer premise equipment (CPE) protocol overhead and B-RAS protocol overhead.

The available range is –120 through 124 bytes. The system rounds up the byte adjustment value to the nearest multiple of 4. For example, a value of 6 is rounded to 8, and a value of -10 is rounded to -8.

RELATED DOCUMENTATION

Bandwidth Management for Downstream Traffic in Edge Networks Overview | 115

Example: Configuring Dynamic Shaping Parameters to Account for Overhead in Downstream Traffic Rates | 119

Verifying the Scheduling and Shaping Configuration for Subscriber Access | 72

Example: Configuring Dynamic Shaping Parameters to Account for Overhead in Downstream Traffic Rates

IN THIS SECTION

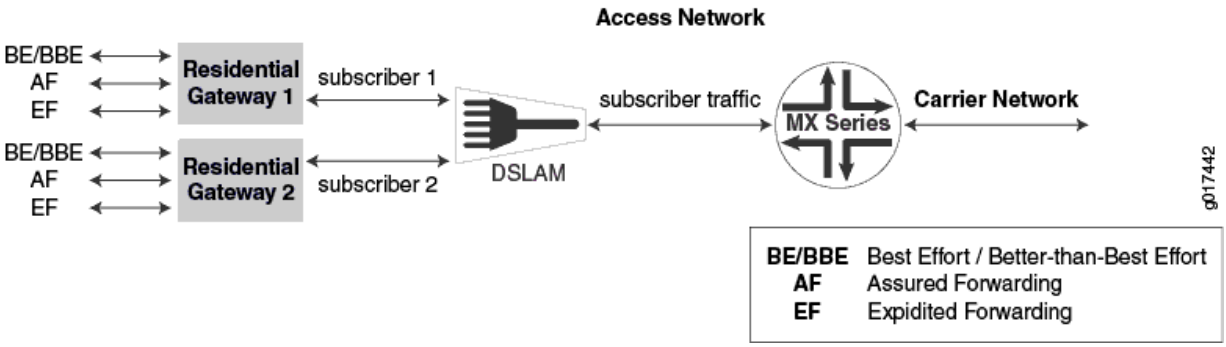
- Managing Traffic with Different Encapsulations | 120
- Managing Downstream Cell-Based Traffic | 122

This topic describes two scenarios for which you can configure dynamic shaping parameters to account for packet overhead in a downstream network.

The RADIUS administrator supplies the initial values on the RADIUS server, and the service activation is performed at subscriber login.

Figure 2 on page 120 shows the sample network that the examples reference.

Figure 2: Sample Network Topology for Downstream Traffic



Managing Traffic with Different Encapsulations

In this example, the MX Series router shown in Figure 2 on page 120 sends stacked VLAN frames to the DSLAM, and the DSLAM sends single-tagged VLAN frames to the residential gateway.

To accurately shape traffic at the residential gateway, the MX Series router must account for the different frame sizes. The difference between the stacked VLAN (S-VLAN) frames sent by the router and the single-tagged VLAN frames received at the residential gateway is a 4-byte VLAN tag. The residential gateway receives frames that are 4 bytes less.

To account for the different frame sizes, you configure the frame shaping mode with -4 byte adjustment:

1. Configure the traffic shaping parameters in the dynamic profile and attach them to the interface.

   Enabling the overhead accounting feature affects the resulting shaping rate, guaranteed rate, and excess rate parameters, if they are configured.

```
[edit]
dynamic-profiles {
    ethernet-downstream-network {
    interfaces {
        $junos-interface-ifd-name {
            unit $junos-underlying-interface-unit {
                family inet;
            }
        }
    }
    class-of-service {
        traffic-control-profiles {
```

```
            tcp-example-overhead-accounting-frame-mode {
                excess-rate percent $junos-cos-excess-rate
                guaranteed-rate $junos-cos-guaranteed-rate
                overhead-accounting $junos-cos-shaping-mode bytes $junos-cos-byte-adjust
                shaping-rate $junos-cos-shaping-rate;
            }
        }
        interfaces {
            $junos-interface-ifd-name {
                unit "$junos-underlying-interface-unit" {
                output-traffic-control-profile tcp1;
                }
            }
        }
    }
    }
}
```

Table 14 on page 121 lists the initial values defined by the RADIUS administrator for the shaping rates.

Table 14: Initial Shaping Values at Subscriber Login For Traffic With Different Encapsulations

Predefined Variable	RADIUS Tag	Value
$junos-cos-shaping-rate	T02	10m
$junos-cos-guaranteed-rate	T03	2m
$junos-cos-excess-rate	T05	50
$junos-cos-shaping-mode	T07	frame-mode
$junos-cos-byte-adjust	T08	–4

2. Verify the adjusted rates.

```
user@host#show class-of-service traffic-control-profile
Traffic control profile: tcp-example-overhead-accounting-frame-mode, Index: 61785
```

```
Excess rate 50
Shaping rate: 10000000
Guaranteed rate: 2000000
Overhead accounting mode: Frame Mode
Overhead bytes: -4
```

Managing Downstream Cell-Based Traffic

In this example, the DSLAM and residential gateway shown in Figure 2 on page 120 are connected through an ATM cell-based network. The MX Series router sends Ethernet frames to the DSLAM, and the DSLAM sends ATM cells to the residential gateway.

To accurately shape traffic at the residential gateway, the MX Series router must account for the different physical network characteristics.

The administrator does not need to configure a byte adjustment value to account for the downstream ATM network, but has the option of configuring a byte adjustment value to account for different encapsulations or decapsulations.

To account for the different frame sizes, configure cell shaping mode:

1. Configure the traffic shaping parameters in the dynamic profile and attach them to the interface.

   Enabling the overhead accounting feature affects the resulting shaping rate, guaranteed rate, and excess rate parameters, if they are configured.

```
[edit]
dynamic-profiles {
    atm-downstream-network {
    interfaces {
        $junos-interface-ifd-name {
            unit $junos-underlying-interface-unit {
                family inet;
            }
        }
    }
    class-of-service {
        traffic-control-profiles {
            tcp-example-overhead-accounting-cell-mode {
                excess-rate percent $junos-cos-excess-rate
                guaranteed-rate $junos-cos-guaranteed-rate
                overhead-accounting $junos-cos-shaping-mode
                shaping-rate $junos-cos-shaping-rate
```

```
            }
        }
        interfaces {
            $junos-interface-ifd-name {
                unit "$junos-underlying-interface-unit" {
                output-traffic-control-profile tcp1;
                }
            }
        }
    }
    }
}
```

Table 15 on page 123 lists the initial values defined by the RADIUS administrator for the shaping rates.

Table 15: Initial Shaping Values at Subscriber Login For Downstream Cell-Based Traffic

Predefined Variable	RADIUS Tag	Value
$junos-cos-shaping-rate	T02	10m
$junos-cos-guaranteed-rate	T03	2m
$junos-cos-excess-rate	T05	50
$junos-cos-shaping-mode	T07	cell-mode

2. Verify the adjusted rates.

```
user@host#show class-of-service traffic-control-profile
Traffic control profile: tcp-example-overhead-accounting-cell-mode, Index: 61785
Shaping rate: 10000000
Excess rate 50
Guaranteed rate: 2000000
Overhead accounting Cell Mode
Overhead bytes: 0
```

To account for ATM segmentation, the MX Series router adjusts all of the rates by 48/53 to account for ATM AAL5 encapsulation. In addition, the router accounts for cell padding, and internally adjusts each frame by 8 bytes to account for the ATM trailer.

RELATED DOCUMENTATION

Configuring Dynamic Shaping Parameters to Account for Overhead in Downstream Traffic Rates | 118

Configuring Static Shaping Parameters to Account for Overhead in Downstream Traffic Rates

The overhead accounting feature enables you to account for downstream traffic that has different encapsulations or downstream traffic from cell-based equipment, such as ATM switches.

You can configure the overhead accounting feature to shape downstream traffic based on frames or cell shaping mode.

You can also account for the different byte sizes per encapsulation by configuring a byte adjustment value for the shaping mode.

To configure the shaping mode and byte adjustment value for static CoS configurations:

1. Specify the shaping mode.

   Frame shaping mode is enabled by default.

```
[edit class-of-service traffic-control-profiles profile-name]
user@host# set overhead-accounting (frame-mode | cell-mode)
```

2. (Optional) Specify a byte adjustment value.

```
[edit class-of-service traffic-control-profiles profile-name
user@host# set overhead-accounting bytes byte-value]
```

BEST PRACTICE: We recommend that you specify a byte adjustment value that represents the difference between the customer premise equipment (CPE) protocol overhead and the B-RAS protocol overhead.

The available range is –120 through 124 bytes. The system rounds up the byte adjustment value to the nearest multiple of 4. For example, a value of 6 is rounded to 8, and a value of –10 is rounded to –8.

RELATED DOCUMENTATION

Bandwidth Management for Downstream Traffic in Edge Networks Overview | 115

Example: Configuring Static Shaping Parameters to Account for Overhead in Downstream Traffic Rates

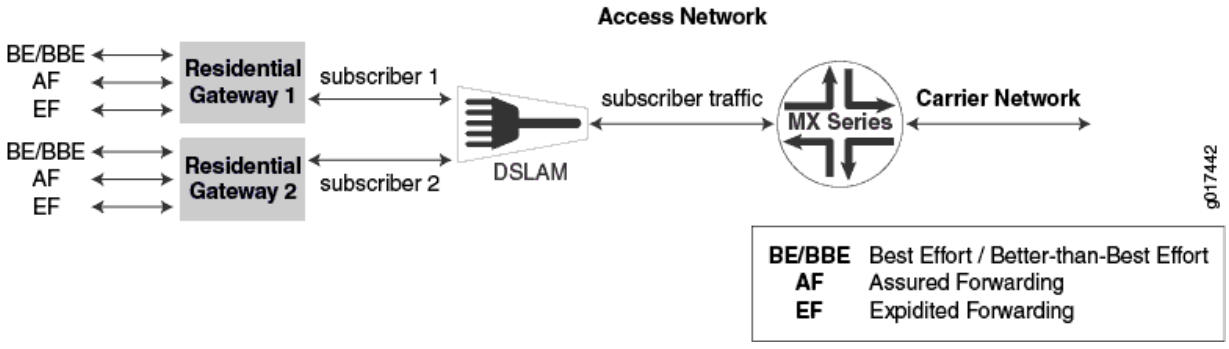
IN THIS SECTION

- Managing Traffic with Different Encapsulations | 126
- Managing Downstream Cell-Based Traffic | 127

This topic describes two scenarios for which you can configure static shaping parameters to account for packet overhead in a downstream network.

Figure 3 on page 125 shows the sample network that the examples reference.

Figure 3: Sample Network Topology for Downstream Traffic



Managing Traffic with Different Encapsulations

In this example, the MX Series router shown in Figure 3 on page 125 sends stacked VLAN frames to the DSLAM, and the DSLAM sends single-tagged VLAN frames to the residential gateway.

To accurately shape traffic at the residential gateway, the MX Series router must account for the different frame sizes. The difference between the stacked VLAN (S-VLAN) frames sent by the router and the single-tagged VLAN frames received at the residential gateway is a 4-byte VLAN tag. The residential gateway receives frames that are 4 bytes less.

To account for the different frame sizes, the network administrator configures the frame shaping mode with –4 byte adjustment:

1. The network administrator configure the traffic shaping parameters and attaches them to the interface.

   Enabling the overhead accounting feature affects the resulting shaping rate, guaranteed rate, and excess rate parameters, if they are configured.

```
[edit]
class-of-service {
    traffic-control-profiles {
        tcp-example-overhead-accounting-frame-mode {
            shaping-rate 10m;
            shaping-rate-priority-high 4m;
            guaranteed-rate 2m;
            excess-rate percent 50;
            overhead-accounting frame-mode bytes -4;
            }
        }
        interfaces {
            ge-1/0/0 {
                output-traffic-control-profile tcp-example-overhead-accounting-frame-mode;
            }
        }
    }
}
```

2. The network administrator verifies the adjusted rates.

```
user@host#show class-of-service traffic-control-profile
Traffic control profile: tcp-example-overhead-accounting-frame-mode, Index: 61785
Shaping rate: 10000000
```

```
Shaping rate priority high: 4000000
Excess rate 50
Guaranteed rate: 2000000
Overhead accounting mode: Frame Mode
Overhead bytes: -4
```

Managing Downstream Cell-Based Traffic

In this example, the DSLAM and residential gateway shown in Figure 3 on page 125 are connected through an ATM cell-based network. The MX Series router sends Ethernet frames to the DSLAM, and the DSLAM sends ATM cells to the residential gateway.

To accurately shape traffic at the residential gateway, the MX Series router must account for the different physical network characteristics.

To account for the different frame sizes, the network administrator configures the cell shaping mode with –4 byte adjustment:

1. Configure the traffic shaping parameters and attach them to the interface.

   Enabling the overhead accounting feature affects the resulting shaping rate, guaranteed rate, and excess rate parameters, if they are configured.

```
[edit]
class-of-service {
    traffic-control-profiles {
        tcp-example-overhead-accounting-cell-mode {
            shaping-rate 10m;
            shaping-rate-priority-high 4m;
            guaranteed-rate 2m;
            excess-rate percent 50;
            overhead-accounting cell-mode;
            }
        }
        interfaces {
            ge-1/0/0 {
                output-traffic-control-profile tcp-example-overhead-accounting-cell-mode;
            }
        }
    }
}
```

2. Verify the adjusted rates.

```
user@host#show class-of-service traffic-control-profile
Traffic control profile: tcp-example-overhead-accounting-cell-mode, Index: 61785
Shaping rate: 10000000
Shaping rate priority high: 4000000
Excess rate 50
Guaranteed rate: 2000000
Overhead accounting mode: Cell Mode
Overhead bytes: 0
```

To account for ATM segmentation, the MX Series router adjusts all of the rates by 48/53 to account for ATM AAL5 encapsulation. In addition, the router accounts for cell padding, and internally adjusts each frame by 8 bytes to account for the ATM trailer.

RELATED DOCUMENTATION

Configuring Static Shaping Parameters to Account for Overhead in Downstream Traffic Rates | 124

Setting Shaping Rate and Overhead Accounting Based on PPPoE Vendor-Specific Tags

IN THIS SECTION

- CLI Interaction with PPPoE Vendor-Specific Tags | 129
- RADIUS Interaction with PPPoE Vendor-Specific Tags | 129
- ANCP Interaction with PPPoE Vendor-Specific Tags | 129
- Multicast QoS Adjustment Interaction with PPPoE Vendor-Specific Tags | 129
- Shaping Rate Restrictions | 130

You can use access line parameters received in PPPoE discovery packets to set the shaping-rate and overhead-accounting class-of-service attributes on dynamic subscriber interfaces in a broadband access network. This feature is supported on MPC/MIC interfaces on MX Series routers.

The shaping rate is based on the Actual-Data-Rate-Downstream attribute.

The overhead accounting value is based on the Access-Loop-Encapsulation attribute and specifies whether the access loop uses Ethernet (frame mode) or ATM (cell mode).

You can configure class-of-service attributes, for example the shaping-rate, using the CLI, RADIUS vendor-specific attributes, ANCP, multicast, or in this case, PPPoE vendor-specific tags.

CLI Interaction with PPPoE Vendor-Specific Tags

When you enable this feature, the values supplied by the PPPoE vendor-specific tags override the parameters that you have configured in the CLI for the `shaping-rate` and `overhead-accounting` statements at the `[edit dynamic-profiles` *`profile-name`* `class-of-service traffic-control-profiles]` hierarchy level. The shaping rate is based on the actual-data-rate-downstream attribute, and is only overridden if the vs-tag value is less than the configured value.

To enable this feature, include the `dynamic-class-of-service-options` statement at the `[edit dynamic-profiles` *`profile-name`* `class-of-service]` hierarchy level. Specify the appropriate attribute as a value for the `vendor-specific-tags` option.

RADIUS Interaction with PPPoE Vendor-Specific Tags

When you enable this feature, the PPPoE vendor-specific tags override the dynamic configuration of the shaping-rate and overhead-accounting values in RADIUS vendor-specific attributes. The shaping-rate value is only overridden if the vs-tag value is less than the RADIUS value.

RADIUS CoA can overwrite the existing values. Upon receipt of a RADIUS CoA, the RADIUS value overrides the value set from the PPPoE vendor-specific tags.

PPPoE vendor-specific tags can override the RADIUS values, but a later RADIUS CoA request can then override that value.

ANCP Interaction with PPPoE Vendor-Specific Tags

You can mix ANCP and PPPoE vendor-specific tags on dynamic PPPoE interfaces, dynamically instantiated PPPoE interfaces, and ACI-sets. ANCP values override the PPPoE values. In this case, the ANCP shaping rate value overrides the PPPoE value.

Multicast QoS Adjustment Interaction with PPPoE Vendor-Specific Tags

Multicast QoS adjustments are not affected by this feature. The multicast adjustments adjust the shaping-rate set by PPPoE vendor-specific tags.

Shaping Rate Restrictions

Shaping rate has the following restrictions regarding the downstream-rate:

- If the downstream-rate is less than the configured shaping-rate (as set in the CLI or using RADIUS attributes) then it is applied, subject to other restrictions. If the downstream-rate is greater than or equal to the configured shaping-rate, no changes are performed.
- The downstream-rate cannot be less than a configured guaranteed-rate. If it is, the downstream-rate is set to the guaranteed-rate.
- The downstream-rate cannot be less than a configured adjust-minimum-rate. If it is, the downstream-rate is set to the adjust-mimimum-rate.
- The downstream-rate cannot be less than 1000 bps. If it is, the downstream-rate is set to 1000 bps.
- The downstream-rate cannot be less than the sum of the transmit-rates of all queues.

RELATED DOCUMENTATION

Bandwidth Management for Downstream Traffic in Edge Networks Overview | 115

Configuring the Shaping Rate and Overhead Accounting Based on PPPoE Vendor-Specific Tags on Dynamic Subscriber Interfaces | 130

Configuring the Shaping Rate and Overhead Accounting Based on PPPoE Vendor-Specific Tags on Dynamic Subscriber Interfaces

To configure the PPPoE vendor-specific tags feature in a dynamic profile:

NOTE: When you enable this feature, the values supplied by the PPPoE vendor-specific tags override the parameters that you have configured for shaping-rate and overhead-accounting statements at the `[edit dynamic-profiles profile-name class-of-service traffic-control-profile]` hierarchy level.

1. (Optional) To configure the shaping rate based on access line information:

```
[edit dynamic-profiles profile-name class-of-service dynamic-class-of-service-options]
user@host# set vendor-specific-tags actual-data-rate-downstream
```

2. (Optional)To configure the overhead-accounting based on access-line information:

```
[edit dynamic-profiles profile-name class-of-service dynamic-class-of-service-options]
user@host# set vendor-specific-tags access-loop-encapsulation
```

RELATED DOCUMENTATION

Setting Shaping Rate and Overhead Accounting Based on PPPoE Vendor-Specific Tags | 128

Bandwidth Management for Downstream Traffic in Edge Networks Overview | 115

Reporting the Effective Shaping Rate for Subscribers

The Effective-Shaping-Rate VSA [26–177] provides the best estimate for a subscriber's downstream traffic rate for accounting purposes. The VSA is included in RADIUS Acct-Start, Acct-Stop, and Interim-Acct messages. The reported rate is the rate enforced on the L3, L2, or L1 node according to local policy. The value of the VSA varies depending on your configuration:

- Actual rate—When effective shaping rate reporting is enabled.
- Advisory rate—When the advisory rate is configured and effective shaping rate reporting is not enabled.
- Port speed—When the advisory rate is not configured and effective shaping rate reporting is not enabled.

When you disable reporting, the VSA reports either the advisory rate or port speed for both existing subscribers and new subscribers that log in after reporting is disabled.

To enable reporting of the actual downstream traffic rate:

- Enable reporting.

```
[edit chassis]
user@host1# set effective-shaping-rate
```

NOTE: When the traffic control profile for the subscriber specifies `cell-mode`, the effective shaping rate does not account for cell padding according to the encapsulation type. The rate includes the 48/53 cell tax.

RELATED DOCUMENTATION

- Verifying the Effective Shaping Rate Reporting Configuration | 132
- Hierarchical CoS Shaping-Rate Adjustments Overview
- Bandwidth Management for Downstream Traffic in Edge Networks Overview | 115
- *Juniper Networks VSAs Supported by the AAA Service Framework*
- *AAA Accounting Messages and Supported RADIUS Attributes and Juniper Networks VSAs for Junos OS*

Verifying the Effective Shaping Rate Reporting Configuration

IN THIS SECTION

- Purpose | 132
- Action | 132

Purpose

Verify whether reporting is enabled for the effective shaping rate. Display the effective shaping rate when reporting is enabled.

Action

- To display configuration information for effective shaping rate reporting:

```
[edit]
user@host# show chassis
...
```

```
effective-shaping-rate;
...
```

- To display the effective shaping rate in kilobits per second when reporting is enabled:

  show subscribers extensive

```
user@host> show subscribers extensive
Type: VLAN
Logical System: default
Routing Instance: default
Interface: demux0.1073741837
Interface type: Dynamic
Interface Set: ifset-1
Underlying Interface: ae1
Dynamic Profile Name: svlan-dhcp-test
State: Active
Session ID: 1
Stacked VLAN Id: 0x8100.201
VLAN Id: 0x8100.201
Login Time: 2011-11-30 00:18:04 PST
Effective shaping-rate: 31000000
...
```

RELATED DOCUMENTATION

Reporting the Effective Shaping Rate for Subscribers | 131

CHAPTER 8

Applying CoS to Households or Individual Subscribers Using ACI-Based Dynamic VLANs

IN THIS CHAPTER

- Applying CoS Attributes to VLANs Using Agent-Circuit-Identifiers | **134**
- Agent Circuit Identifier-Based Dynamic VLANs Bandwidth Management Overview | **137**
- Restrictions for Configuring Adjustment of CoS Shaping Rate and Overhead Accounting for Dynamic ACI Interface Sets | **141**
- Adjusting the CoS Shaping Rate and Overhead Accounting Parameters for Agent Circuit Identifier-Based Dynamic VLANs | **142**

Applying CoS Attributes to VLANs Using Agent-Circuit-Identifiers

To apply CoS attributes, such as shaping, at the household level, you must set and define the CoS policy for the ACI VLAN interface set using the dynamic profile for the ACI interface set (not the subscriber profile). You can also configure a traffic-control profile and a remaining traffic-control profile for a dynamic interface set.

The following example is a CoS profile for an ACI interface set using a unique-ID based dynamic scheduler map:

Configure a CoS dynamic profile with a simple traffic-control profile that is applied to the dynamic interface set that represents the ACI VLAN.

1. Configure CoS to support a dynamic interface set in the CoS profile:

```
[edit dynamic-profiles profile-name]
user@host# edit interface "$junos-interface-name"
```

2. Configure the interfaces.

```
[edit dynamic-profiles profile-name interfaces]
user@host# edit interface-set "$junos-interface-set-name"
user@host# edit interface "$junos-interface-ifd-name"
```

3. Configure the CoS traffic-control profile.

```
[edit class-of-service]
user@host# edit traffic-control-profiles traffic-control-profile-name
user@host# set shaping-rate rate
user@host# set guaranteed-rate rate
```

4. Specify the interfaces.

```
[edit class-of-service interfaces]
user@host# edit interface-set "$junos-interface-set-name"
user@host# edit output-traffic-control-profile profile-name
```

The following example is a CoS profile for an ACI set using a unique ID-based dynamic scheduler map:

```
dynamic-profiles {
    aci-set-profile {
    variables {
        ds1q0q2DP uid;
        ds1q1q2DP uid;
        be1_dp uid;
        ef1_dp uid;
        af1_dp uid;
        nc1_dp uid;
    }
    interfaces {
        interface-set "$junos-interface-set-name" {
            interface "$junos-interface-ifd-name";
        }
    }
    class-of-service {
        traffic-control-profiles {
            tcp2 {
                scheduler-map ss1q0q1DP;
```

```
            shaping-rate 50m;
            guaranteed-rate 30m;
            overhead-accounting bytes -20;
        }
        tcp3 {
            scheduler-map "$ds1q1q2DP";
            shaping-rate 30m;
            guaranteed-rate 10m;
            overhead-accounting bytes -20;
        }
    }
    interfaces {
        interface-set "$junos-interface-set-name" {
            output-traffic-control-profile tcp2;
            output-traffic-control-profile-remaining tcp3;
        }
    }
    scheduler-maps {
        "$ds1q0q2DP" {
            forwarding-class be scheduler "$be1_dp";
            forwarding-class af scheduler "$af1_dp";
            forwarding-class nc scheduler "$nc1_dp";
        }
        "$ds1q1q2DP" {
            forwarding-class ef scheduler "$ef1_dp";
            forwarding-class af scheduler "$af1_dp";
            forwarding-class nc scheduler "$nc1_dp";
        }
    }
    schedulers {
        "$be1_dp" {
            transmit-rate percent 25;
            priority low;
            drop-profile-map loss-priority low protocol any drop-profile d3;
            drop-profile-map loss-priority medium-low protocol any drop-profile d2;
            drop-profile-map loss-priority medium-high protocol any drop-profile d1;
            drop-profile-map loss-priority high protocol any drop-profile d0;
        }
        "$ef1_dp" {
            transmit-rate percent 25;
            priority low;
            drop-profile-map loss-priority low protocol any drop-profile d3;
            drop-profile-map loss-priority medium-low protocol any drop-profile d2;
```

```
                drop-profile-map loss-priority medium-high protocol any drop-profile d1;
                drop-profile-map loss-priority high protocol any drop-profile d0;
            }
            "$af1_dp" {
                transmit-rate percent 25;
                priority low;
                drop-profile-map loss-priority low protocol any drop-profile d3;
                drop-profile-map loss-priority medium-low protocol any drop-profile d2;
                drop-profile-map loss-priority medium-high protocol any drop-profile d1;
                drop-profile-map loss-priority high protocol any drop-profile d0;
            }
            "$nc1_dp" {
                transmit-rate percent 25;
                priority low;
                drop-profile-map loss-priority low protocol any drop-profile d3;
                drop-profile-map loss-priority medium-low protocol any drop-profile d2;
                drop-profile-map loss-priority medium-high protocol any drop-profile d1;
                drop-profile-map loss-priority high protocol any drop-profile d0;
            }
        }
    }
}
```

RELATED DOCUMENTATION

Guidelines for Configuring Dynamic CoS for Subscriber Access | 44

Changing CoS Services Overview | 166

Agent Circuit Identifier-Based Dynamic VLANs Bandwidth Management Overview

IN THIS SECTION

- CoS Shaping Rate Adjustment | 138
- CoS Overhead Accounting Adjustment | 139

Dynamic Profiles and Adjustment of CoS Shaping Rate and Overhead Accounting | 139
Guidelines for Configuring Adjustment of CoS Shaping Rate and Overhead Accounting | 140

A router in a subscriber access network ensures *class of service* (CoS) for dynamic subscriber interfaces. An MX Series router with Modular Port Concentrator/Modular Interface Card (MPC/MIC) interfaces ensures that subscribers receive an adequate minimum bandwidth, referred to as the *guaranteed rate*, and maximum bandwidth, referred to as the *shaping rate*. For dynamic VLAN subscriber interfaces based on agent circuit identifier (ACI) information, you can shape the bandwidth either at a per-household level for a dynamic ACI interface set, or at a per-subscriber level for a dynamic VLAN subscriber interface associated with an ACI interface set.

To help you manage bandwidth more efficiently and economically for ACI-based dynamic VLAN subscriber interfaces for PPPoE subscribers, you can configure the router to use specific PPPoE vendor-specific attributes (VSAs) found in PPPoE control packets to adjust the CoS shaping-rate and overhead-accounting attributes for dynamic ACI interface sets and their associated ACI-based dynamic VLAN subscriber interfaces.

This overview covers the following topics:

CoS Shaping Rate Adjustment

The CoS shaping rate adjustment is based on the value of the Actual-Data-Rate-Downstream DSL Forum VSA [26-130] found in PPPoE Active Discovery Initiation (PADI) and PPPoE Active Discovery Request (PADR) control packets for PPPoE traffic. The Actual-Data-Rate-Downstream VSA contains the actual downstream data rate, in kilobits per second, of the subscriber's synchronized digital subscriber line (DSL) link.

To configure the router to use the Actual-Data-Rate-Downstream VSA to adjust the CoS shaping-rate attribute, include the `vendor-specific-tags` statement with the `actual-data-rate-downstream` option at the `[edit dynamic-profiles` *`profile-name`* `class-of-service dynamic-class-of-service-options]` hierarchy level in either the dynamic profile that defines the ACI interface set or the dynamic profile that configures the associated dynamic PPPoE (`pp0`) subscriber interface.

When you enable this feature, the value of the Actual-Data-Rate-Downstream VSA overrides the `shaping-rate` value configured at the `[edit dynamic-profiles` *`profile-name`* `class-of-service traffic-control-profiles]` hierarchy level only if the Actual-Data-Rate-Downstream VSA value is less than the `shaping-rate` value configured with the CLI.

CoS Overhead Accounting Adjustment

The CoS overhead accounting adjustment is based on the value of the Access-Loop-Encapsulation DSL Forum VSA [26-144] found in PADI and PADR control packets for PPPoE traffic. The Access-Loop-Encapsulation VSA identifies the encapsulation used by the subscriber associated with the digital subscriber line access multiplexer (DSLAM) access loop from which requests are initiated.

The value of the Data Link subfield in the Access-Loop-Encapsulation VSA determines the overhead accounting mode in use on the access loop. If the Data Link subfield value is 0 (ATM Adaptation Layer 5, or AAL5), the access loop uses cell-mode encapsulation. If the Data Link subfield value is 1 (Ethernet), the access loop uses frame-mode encapsulation.

In subscriber access networks where the router passes downstream ATM traffic to Ethernet interfaces, the different Layer 2 encapsulations between the router and the PPPoE Intermediate Agent on the DSLAM make managing the bandwidth of downstream ATM traffic difficult. Using the Access-Loop-Encapsulation VSA to shape traffic based on frames or cells enables the router to adjust the overhead-accounting attribute in order to apply the correct downstream rate for the subscriber.

To configure the router to use the Access-Loop-Encapsulation VSA to adjust the CoS overhead-accounting attribute, include the `vendor-specific-tags` statement with the `access-loop-encapsulation` option at the `[edit dynamic-profiles` *`profile-name`* `class-of-service dynamic-class-of-service-options]` hierarchy level in either the dynamic profile that defines the ACI interface set or the dynamic profile that configures the associated dynamic PPPoE (`pp0`) subscriber interface.

When you enable this feature, the value of the Access-Loop-Encapsulation VSA always overrides the `overhead-accounting` value configured at the `[edit dynamic-profiles` *`profile-name`* `class-of-service traffic-control-profiles]` hierarchy level.

Dynamic Profiles and Adjustment of CoS Shaping Rate and Overhead Accounting

When you configure the router to use one or both of the Actual-Data-Rate-Downstream VSA value and Access-Loop-Encapsulation VSA value to adjust the CoS shaping rate and overhead accounting attributes, respectively, the router adjusts these attributes when the dynamic ACI interface set is created and the router receives the PADI and PADR packets from the first subscriber interface belonging to the ACI interface set.

You can configure CoS adjustment based on either or both VSAs in either or both of the following dynamic profiles:

- To configure adjustment of the CoS shaping rate and overhead accounting on a per-household basis, use the dynamic profile that defines the dynamic ACI interface set.
- To configure adjustment of the CoS shaping rate and overhead accounting on a per-subscriber basis, use the dynamic profile that defines the ACI-based dynamic PPPoE (`pp0`) subscriber interface associated with the ACI interface set.

Table 16 on page 140 summarizes how the dynamic profile in which you configure CoS adjustment for ACI-based dynamic VLANs using one or both VSAs affects the router behavior.

Table 16: CoS Adjustment in Dynamic Profiles for ACI Interface Sets and ACI-Based Subscriber Interfaces

VSAs Specified in ACI Interface Set Dynamic Profile	VSAs Specified in PPPoE Subscriber Interface Dynamic Profile	Result
Yes	No	Router adjusts specified CoS attributes only for dynamic ACI interface set
No	Yes	Router adjusts specified CoS attributes only for ACI-based dynamic PPPoE subscriber interface
Yes	Yes	Router adjusts specified CoS attributes for both dynamic ACI interface set and ACI-based dynamic PPPoE subscriber interface
No	No	Router does not adjust CoS attributes for either the dynamic ACI interface set or the ACI-based dynamic PPPoE subscriber interface

Guidelines for Configuring Adjustment of CoS Shaping Rate and Overhead Accounting

You can also configure the router to use the Actual-Data-Rate-Downstream VSA and Access-Loop-Encapsulation VSA values in PPPoE control packets to adjust the CoS shaping rate and overhead accounting attributes, respectively, for dynamic subscriber interfaces *not* associated with dynamic ACI interface sets.

With the exception of the constraints described in "Restrictions for Configuring Adjustment of CoS Shaping Rate and Overhead Accounting for Dynamic ACI Interface Sets" on page 141, most of the guidelines and restrictions that apply to this feature for use with non–ACI-based dynamic subscriber interfaces also apply to its use for dynamic ACI interface sets and their associated ACI-based dynamic VLAN subscriber interfaces.

RELATED DOCUMENTATION

Setting Shaping Rate and Overhead Accounting Based on PPPoE Vendor-Specific Tags | 128

Adjusting the CoS Shaping Rate and Overhead Accounting Parameters for Agent Circuit Identifier-Based Dynamic VLANs | 142

Restrictions for Configuring Adjustment of CoS Shaping Rate and Overhead Accounting for Dynamic ACI Interface Sets | 141

Restrictions for Configuring Adjustment of CoS Shaping Rate and Overhead Accounting for Dynamic ACI Interface Sets

The following restrictions apply when you configure the router to use the Actual-Data-Rate-Downstream VSA and Access-Loop-Encapsulation vendor-specific attribute (VSA) values in PPPoE control packets to adjust the CoS shaping rate and overhead accounting attributes, respectively, for dynamic ACI interface sets and their associated agent circuit identifier (ACI)-based dynamic VLAN subscriber interfaces:

- You cannot configure adjustment of CoS shaping rate and overhead accounting attributes based on Actual-Data-Rate-Downstream VSA and Access-Loop-Encapsulation VSA values that the router receives from the following sources:
  - RADIUS servers
  - Access Node Control Protocol (ANCP) access loop information
  - Dynamic Host Configuration Protocol (DHCP) discovery packets
- You cannot use this feature to report information about the PPPoE VSA values to RADIUS.
- You cannot use this feature to configure CoS adjustment of upstream data traffic on a dynamic ACI interface set.

RELATED DOCUMENTATION

Agent Circuit Identifier-Based Dynamic VLANs Bandwidth Management Overview | 137

Setting Shaping Rate and Overhead Accounting Based on PPPoE Vendor-Specific Tags | 128

Adjusting the CoS Shaping Rate and Overhead Accounting Parameters for Agent Circuit Identifier-Based Dynamic VLANs | 142

Adjusting the CoS Shaping Rate and Overhead Accounting Parameters for Agent Circuit Identifier-Based Dynamic VLANs

You can configure the router to use either or both of the Actual-Data-Rate-Downstream [26-130] or Access-Loop-Encapsulation [26-144] DSL Forum vendor-specific attribute (VSA) values in PPPoE control packets to adjust the CoS shaping-rate and overhead-accounting attributes, respectively, for dynamic agent circuit identifier (ACI) interface sets and their associated ACI-based dynamic VLAN subscriber interfaces.

Before you begin:

- To configure adjustment of the CoS shaping rate and overhead accounting attributes on a per-household basis, create a dynamic profile that defines the dynamic ACI interface set.

  See *Defining ACI Interface Sets*.

- To configure adjustment of the CoS shaping rate and overhead accounting attributes on a per-subscriber basis, create a dynamic profile that defines the ACI-based dynamic PPPoE (`pp0`) subscriber interface associated with the ACI interface set.

  See *Configuring Dynamic VLAN Subscriber Interfaces Based on Agent Circuit Identifier Information*.

To configure the router to use the Actual-Data-Rate-Downstream or Access-Loop-Encapsulation VSA values in PPPoE control packets to adjust the CoS shaping-rate and overhead-accounting attributes for dynamic ACI interface sets and associated ACI-based dynamic VLAN subscriber interfaces, do either or both of the following:

- In a dynamic profile for an ACI interface set or a dynamic profile for an ACI-based PPPoE subscriber interface, configure adjustment of the CoS shaping-rate attribute based on the value of the Actual-Data-Rate-Downstream VSA.

```
[edit dynamic-profiles profile-name class-of-service dynamic-class-of-service-options]
user@host# set vendor-specific-tags actual-data-rate-downstream
```

- In a dynamic profile for an ACI interface set or a dynamic profile for an ACI-based PPPoE subscriber interface, configure adjustment of the CoS overhead-accounting attribute based on the value of the Access-Loop-Encapsulation VSA.

```
[edit dynamic-profiles profile-name class-of-service dynamic-class-of-service-options]
user@host# set vendor-specific-tags access-loop-encapsulation
```

RELATED DOCUMENTATION

Agent Circuit Identifier-Based Dynamic VLANs Bandwidth Management Overview | **137**

Restrictions for Configuring Adjustment of CoS Shaping Rate and Overhead Accounting for Dynamic ACI Interface Sets | **141**

Configuring Dynamic VLANs Based on Agent Circuit Identifier Information

CHAPTER 9

Applying CoS to Households or Individual Subscribers Using Access Line Identifier Dynamic VLANs

IN THIS CHAPTER

- Applying CoS Attributes to VLANs Using Access-Line Identifiers | 144
- Bandwidth Management Overview for Dynamic VLANs Based on Access-Line Identifiers | 147
- Restrictions for Configuring Adjustment of CoS Shaping Rate and Overhead Accounting for Dynamic ALI Interface Sets | 151
- Adjusting the CoS Shaping Rate and Overhead Accounting Parameters for Dynamic VLANs Based on Access-Line Identifiers | 152

Applying CoS Attributes to VLANs Using Access-Line Identifiers

To apply CoS attributes, such as shaping, at the household level, you must set and define the CoS policy for the access-line-identifier (ALI) VLAN interface set using the dynamic profile for the ALI interface set (not the subscriber profile). You can also configure a traffic-control profile and a remaining traffic-control profile for a dynamic interface set.

The following example is a CoS profile for an ALI interface set using a unique-ID based dynamic scheduler map:

Configure a CoS dynamic profile with a simple traffic-control profile that is applied to the dynamic interface set that represents the ALI VLAN.

1. Configure CoS to support a dynamic interface set in the CoS profile:

```
[edit dynamic-profiles profile-name]
user@host# edit interface "$junos-interface-name"
```

2. Configure the interfaces.

```
[edit dynamic-profiles profile-name interfaces]
user@host# edit interface-set "$junos-interface-set-name"
user@host# edit interface "$junos-interface-ifd-name"
```

3. Configure the CoS traffic-control profile.

```
[edit class-of-service]
user@host# edit traffic-control-profiles traffic-control-profile-name
user@host# set shaping-rate rate
user@host# set guaranteed-rate rate
```

4. Specify the interfaces.

```
[edit class-of-service interfaces]
user@host# edit interface-set "$junos-interface-set-name"
user@host# edit output-traffic-control-profile profile-name
```

The following example is a CoS profile for an ALI set using a unique ID-based dynamic scheduler map:

```
dynamic-profiles {
    ali-set-profile {
    variables {
        ds1q0q2DP uid;
        ds1q1q2DP uid;
        be1_dp uid;
        ef1_dp uid;
        af1_dp uid;
        nc1_dp uid;
    }
    interfaces {
        interface-set "$junos-interface-set-name" {
            interface "$junos-interface-ifd-name";
        }
    }
    class-of-service {
        traffic-control-profiles {
            tcp2 {
                scheduler-map ss1q0q1DP;
```

```
            shaping-rate 50m;
            guaranteed-rate 30m;
            overhead-accounting bytes -20;
        }
        tcp3 {
            scheduler-map "$ds1q1q2DP";
            shaping-rate 30m;
            guaranteed-rate 10m;
            overhead-accounting bytes -20;
        }
    }
    interfaces {
        interface-set "$junos-interface-set-name" {
            output-traffic-control-profile tcp2;
            output-traffic-control-profile-remaining tcp3;
        }
    }
    scheduler-maps {
        "$ds1q0q2DP" {
            forwarding-class be scheduler "$be1_dp";
            forwarding-class af scheduler "$af1_dp";
            forwarding-class nc scheduler "$nc1_dp";
        }
        "$ds1q1q2DP" {
            forwarding-class ef scheduler "$ef1_dp";
            forwarding-class af scheduler "$af1_dp";
            forwarding-class nc scheduler "$nc1_dp";
        }
    }
    schedulers {
        "$be1_dp" {
            transmit-rate percent 25;
            priority low;
            drop-profile-map loss-priority low protocol any drop-profile d3;
            drop-profile-map loss-priority medium-low protocol any drop-profile d2;
            drop-profile-map loss-priority medium-high protocol any drop-profile d1;
            drop-profile-map loss-priority high protocol any drop-profile d0;
        }
        "$ef1_dp" {
            transmit-rate percent 25;
            priority low;
            drop-profile-map loss-priority low protocol any drop-profile d3;
            drop-profile-map loss-priority medium-low protocol any drop-profile d2;
```

```
                drop-profile-map loss-priority medium-high protocol any drop-profile d1;
                drop-profile-map loss-priority high protocol any drop-profile d0;
            }
            "$af1_dp" {
                transmit-rate percent 25;
                priority low;
                drop-profile-map loss-priority low protocol any drop-profile d3;
                drop-profile-map loss-priority medium-low protocol any drop-profile d2;
                drop-profile-map loss-priority medium-high protocol any drop-profile d1;
                drop-profile-map loss-priority high protocol any drop-profile d0;
            }
            "$nc1_dp" {
                transmit-rate percent 25;
                priority low;
                drop-profile-map loss-priority low protocol any drop-profile d3;
                drop-profile-map loss-priority medium-low protocol any drop-profile d2;
                drop-profile-map loss-priority medium-high protocol any drop-profile d1;
                drop-profile-map loss-priority high protocol any drop-profile d0;
            }
        }
    }
}
```

RELATED DOCUMENTATION

Guidelines for Configuring Dynamic CoS for Subscriber Access | 44

Changing CoS Services Overview | 166

Bandwidth Management Overview for Dynamic VLANs Based on Access-Line Identifiers

IN THIS SECTION

- CoS Shaping Rate Adjustment | 148
- CoS Overhead Accounting Adjustment | 149

Dynamic Profiles and Adjustment of CoS Shaping Rate and Overhead Accounting | 149
Guidelines for Configuring Adjustment of CoS Shaping Rate and Overhead Accounting | 150

A router in a subscriber access network ensures *class of service* (CoS) for dynamic subscriber interfaces. An MX Series router with Modular Port Concentrator/Modular Interface Card (MPC/MIC) interfaces ensures that subscribers receive an adequate minimum bandwidth, referred to as the *guaranteed rate*, and maximum bandwidth, referred to as the *shaping rate*. For dynamic VLAN subscriber interfaces based on access-line identifiers (ALI), you can shape the bandwidth either at a per-household level for a dynamic ALI interface set, or at a per-subscriber level for a dynamic VLAN subscriber interface associated with an ALI interface set.

To help you manage bandwidth efficiently and economically for ALI-based dynamic VLAN subscriber interfaces for PPPoE subscribers, you can configure the router to use specific PPPoE vendor-specific attributes (VSAs) found in PPPoE control packets to adjust the CoS shaping-rate and overhead-accounting attributes for dynamic ALI interface sets and their associated ALI-based dynamic VLAN subscriber interfaces.

This overview covers the following topics:

CoS Shaping Rate Adjustment

The CoS shaping rate adjustment is based on the value of the Actual-Data-Rate-Downstream DSL Forum VSA [26-130] found in PPPoE Active Discovery Initiation (PADI) and PPPoE Active Discovery Request (PADR) control packets for PPPoE traffic. The Actual-Data-Rate-Downstream VSA contains the actual downstream data rate, in bits per second, of the subscriber's synchronized DSL link.

To configure the router to use the Actual-Data-Rate-Downstream VSA to adjust the CoS shaping-rate attribute, include the `vendor-specific-tags` statement with the `actual-data-rate-downstream` option at the `[edit dynamic-profiles` *`profile-name`* `class-of-service dynamic-class-of-service-options]` hierarchy level in either the dynamic profile that defines the ALI interface set or the dynamic profile that configures the associated dynamic PPPoE (`pp0`) subscriber interface.

When you enable this feature, the value of the Actual-Data-Rate-Downstream VSA overrides the `shaping-rate` value configured at the `[edit dynamic-profiles` *`profile-name`* `class-of-service traffic-control-profiles]` hierarchy level only if the Actual-Data-Rate-Downstream VSA value is less than the `shaping-rate` value configured with the CLI.

CoS Overhead Accounting Adjustment

The CoS overhead accounting adjustment is based on the value of the Access-Loop-Encapsulation DSL Forum VSA [26-144] found in PADI and PADR control packets for PPPoE traffic. The Access-Loop-Encapsulation VSA identifies the encapsulation used by the subscriber associated with the DSL access multiplexer (DSLAM) access loop from which requests are initiated.

The value of the Data Link subfield in the Access-Loop-Encapsulation VSA determines the overhead accounting mode in use on the access loop. If the Data Link subfield value is 0 (ATM Adaptation Layer 5, or AAL5), the access loop uses cell-mode encapsulation. If the Data Link subfield value is 1 (Ethernet), the access loop uses frame-mode encapsulation.

In subscriber access networks where the router passes downstream ATM traffic to Ethernet interfaces, the different Layer 2 encapsulations between the router and the PPPoE Intermediate Agent on the DSLAM make managing the bandwidth of downstream ATM traffic difficult. Using the Access-Loop-Encapsulation VSA to shape traffic based on frames or cells enables the router to adjust the overhead-accounting attribute to apply the correct downstream rate for the subscriber.

To configure the router to use the Access-Loop-Encapsulation VSA to adjust the CoS overhead-accounting attribute, include the `vendor-specific-tags` statement with the `access-loop-encapsulation` option at the `[edit dynamic-profiles` *`profile-name`* `class-of-service dynamic-class-of-service-options]` hierarchy level in either the dynamic profile that defines the ALI interface set or the dynamic profile that configures the associated dynamic PPPoE (`pp0`) subscriber interface.

When you enable this feature, the value of the Access-Loop-Encapsulation VSA always overrides the `overhead-accounting` value configured at the `[edit dynamic-profiles` *`profile-name`* `class-of-service traffic-control-profiles]` hierarchy level.

Dynamic Profiles and Adjustment of CoS Shaping Rate and Overhead Accounting

When you configure the router to use either or both of the Actual-Data-Rate-Downstream VSA value and Access-Loop-Encapsulation VSA value to adjust the CoS shaping rate and overhead accounting attributes, respectively, the router adjusts these attributes when the dynamic ALI interface set is created and the router receives the PADI and PADR packets from the first subscriber interface belonging to the ALI interface set.

You can configure CoS adjustment based on either or both VSAs in either or both of the following dynamic profiles:

- To configure adjustment of the CoS shaping rate and overhead accounting on a per-household basis, use the dynamic profile that defines the dynamic ALI interface set.
- To configure adjustment of the CoS shaping rate and overhead accounting on a per-subscriber basis, use the dynamic profile that defines the ALI-based dynamic PPPoE (`pp0`) subscriber interface associated with the ALI interface set.

Table 17 on page 150 summarizes how the dynamic profile in which you configure CoS adjustment for ALI-based dynamic VLANs using one or both VSAs affects the router behavior.

Table 17: CoS Adjustment in Dynamic Profiles for ALI Interface Sets and ALI-Based Subscriber Interfaces

VSAs Specified in ALI Interface Set Dynamic Profile	VSAs Specified in PPPoE Subscriber Interface Dynamic Profile	Result
Yes	No	Router adjusts specified CoS attributes only for dynamic ALI interface set
No	Yes	Router adjusts specified CoS attributes only for ALI-based dynamic PPPoE subscriber interface
Yes	Yes	Router adjusts specified CoS attributes for both dynamic ALI interface set and ALI-based dynamic PPPoE subscriber interface
No	No	Router does not adjust CoS attributes for either the dynamic ALI interface set or the ALI-based dynamic PPPoE subscriber interface

Guidelines for Configuring Adjustment of CoS Shaping Rate and Overhead Accounting

You can also configure the router to use the Actual-Data-Rate-Downstream VSA and Access-Loop-Encapsulation VSA values in PPPoE control packets to adjust the CoS shaping rate and overhead accounting attributes, respectively, for dynamic subscriber interfaces *not* associated with dynamic ALI interface sets.

With the exception of the constraints described in "Restrictions for Configuring Adjustment of CoS Shaping Rate and Overhead Accounting for Dynamic ALI Interface Sets" on page 151, most of the guidelines and restrictions that apply to this feature for use with dynamic subscriber interfaces that are not based on ALIs also apply to its use for dynamic ALI interface sets and their associated ALI-based dynamic VLAN subscriber interfaces.

RELATED DOCUMENTATION

Restrictions for Configuring Adjustment of CoS Shaping Rate and Overhead Accounting for Dynamic ALI Interface Sets | 151

Setting Shaping Rate and Overhead Accounting Based on PPPoE Vendor-Specific Tags | 128

Adjusting the CoS Shaping Rate and Overhead Accounting Parameters for Dynamic VLANs Based on Access-Line Identifiers | 152

Restrictions for Configuring Adjustment of CoS Shaping Rate and Overhead Accounting for Dynamic ALI Interface Sets

The following restrictions apply when you configure the router to use the Actual-Data-Rate-Downstream VSA and Access-Loop-Encapsulation vendor-specific attribute (VSA) values in PPPoE control packets to adjust the CoS shaping rate and overhead accounting attributes, respectively, for dynamic interface sets based on the access-line identifier (ALI) and their associated ALI-based dynamic VLAN subscriber interfaces:

- You cannot configure adjustment of CoS shaping rate and overhead accounting attributes based on Actual-Data-Rate-Downstream VSA and Access-Loop-Encapsulation VSA values that the router receives from the following sources:
  - RADIUS servers
  - Access Node Control Protocol (ANCP) access loop information
  - Dynamic Host Configuration Protocol (DHCP) discovery packets
- You cannot use this feature to report information about the PPPoE VSA values to RADIUS.
- You cannot use this feature to configure CoS adjustment of upstream data traffic on a dynamic ACI interface set.

RELATED DOCUMENTATION

Bandwidth Management Overview for Dynamic VLANs Based on Access-Line Identifiers | 147

Setting Shaping Rate and Overhead Accounting Based on PPPoE Vendor-Specific Tags | 128

Adjusting the CoS Shaping Rate and Overhead Accounting Parameters for Dynamic VLANs Based on Access-Line Identifiers | 152

Adjusting the CoS Shaping Rate and Overhead Accounting Parameters for Dynamic VLANs Based on Access-Line Identifiers

You can configure the router to use either or both of the Actual-Data-Rate-Downstream [26-130] or Access-Loop-Encapsulation [26-144] DSL Forum vendor-specific attribute (VSA) values in PPPoE control packets to adjust the CoS shaping-rate and overhead-accounting attributes, respectively, for dynamic access-line-identifier (ALI) interface sets and their associated ALI-based dynamic VLAN subscriber interfaces.

Before you begin:

- To configure adjustment of the CoS shaping rate and overhead accounting attributes on a per-household basis, create a dynamic profile that defines the dynamic ALI interface set.

  See *Defining Access-Line-Identifier Interface Sets*.

- To configure adjustment of the CoS shaping rate and overhead accounting attributes on a per-subscriber basis, create a dynamic profile that defines the ALI-based dynamic PPPoE (`pp0`) subscriber interface associated with the ALI interface set.

  See *Configuring Dynamic VLAN Subscriber Interfaces Based on Access-Line Identifiers*.

To configure the router to use the Actual-Data-Rate-Downstream or Access-Loop-Encapsulation VSA values in PPPoE control packets to adjust the CoS shaping-rate and overhead-accounting attributes for dynamic ALI interface sets and associated ALI-based dynamic VLAN subscriber interfaces, do either or both of the following:

- In a dynamic profile for an ALI interface set or a dynamic profile for an ALI-based PPPoE subscriber interface, configure adjustment of the CoS shaping-rate attribute based on the value of the Actual-Data-Rate-Downstream VSA.

```
[edit dynamic-profiles profile-name class-of-service dynamic-class-of-service-options]
user@host# set vendor-specific-tags actual-data-rate-downstream
```

- In a dynamic profile for an ALI interface set or a dynamic profile for an ALI-based PPPoE subscriber interface, configure adjustment of the CoS overhead-accounting attribute based on the value of the Access-Loop-Encapsulation VSA.

```
[edit dynamic-profiles profile-name class-of-service dynamic-class-of-service-options]
user@host# set vendor-specific-tags access-loop-encapsulation
```

RELATED DOCUMENTATION

Configuring Dynamic VLANs Based on Access-Line Identifiers

Bandwidth Management Overview for Dynamic VLANs Based on Access-Line Identifiers | **147**

Restrictions for Configuring Adjustment of CoS Shaping Rate and Overhead Accounting for Dynamic ALI Interface Sets | **151**

CHAPTER 10

Managing Excess Bandwidth Distribution and Traffic Bursts

IN THIS CHAPTER

- Excess Bandwidth Distribution on MIC and MPC Interfaces Overview | 154
- Traffic Burst Management on MIC and MPC Interfaces Overview | 155
- Managing Excess Bandwidth Distribution for Dynamic CoS on MIC and MPC Interfaces | 158

Excess Bandwidth Distribution on MIC and MPC Interfaces Overview

Service providers often used tiered services to provide bandwidth for excess traffic as traffic patterns vary. By default, excess bandwidth between a configured guaranteed rate and shaping rate is shared equally among all queues on MIC and MPC interfaces, which might not be optimal for all subscribers to a service.

You can adjust this distribution by configuring the rates and priorities for the excess bandwidth.

By default, when traffic exceeds the shaping or guaranteed rates, the system demotes traffic with guaranteed high (GH) priority and guaranteed medium (GM) priority. You can disable this priority demotion for the MIC and MPC interfaces in your router.

RELATED DOCUMENTATION

- Managing Excess Bandwidth Distribution on Static Interfaces on MICs and MPCs
- Managing Excess Bandwidth Distribution for Dynamic CoS on MIC and MPC Interfaces | 158
- Per-Priority Shaping on MIC and MPC Interfaces Overview
- Traffic Burst Management on MIC and MPC Interfaces Overview | 155

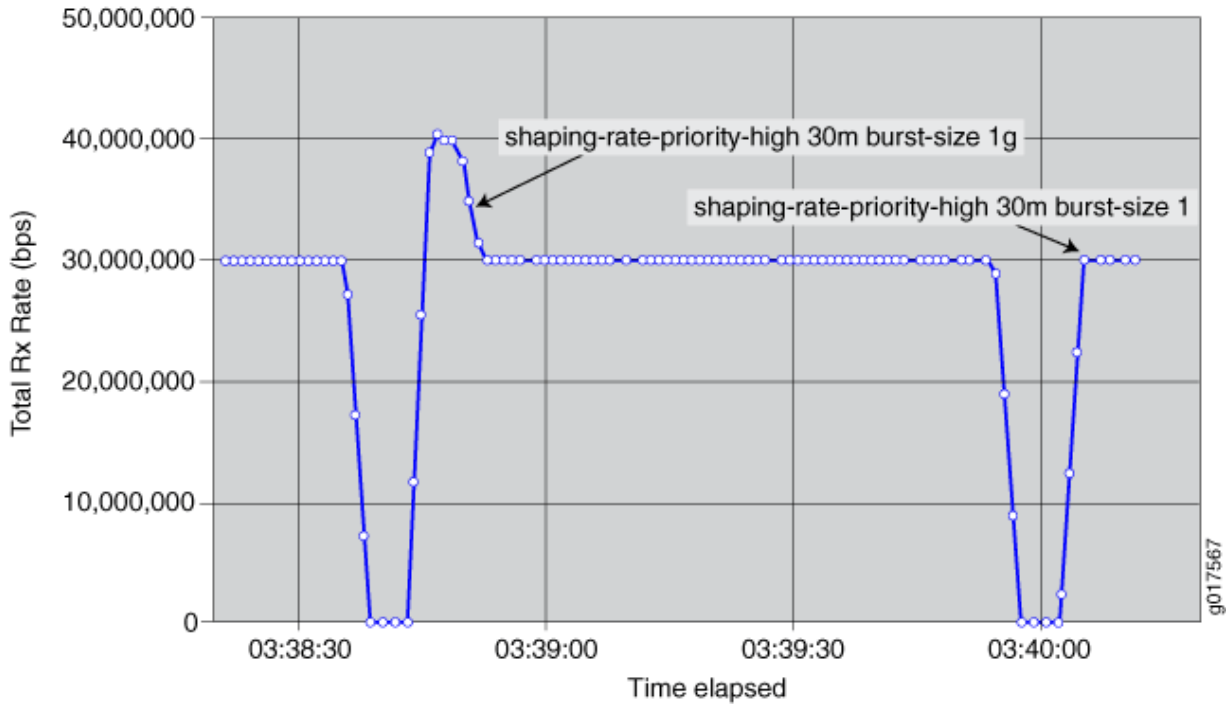
Traffic Burst Management on MIC and MPC Interfaces Overview

IN THIS SECTION

- Guidelines for Configuring the Burst Size | 156
- How the System Calculates the Burst Size | 157

You can manage the impact of bursts of traffic on your network by configuring a burst-size value with the shaping rate or the guaranteed rate. The value is the maximum bytes of rate credit that can accrue for an idle queue or scheduler node. When a queue or node becomes active, the accrued rate credits enable the queue or node to catch up to the configured rate.

Figure 4: Sample Burst Shaping Rates



In Figure 4 on page 155, the network administrator configures a large burst-size value for the shaping rate, then configures a small burst-size value. The larger burst size is subject to a maximum value. The smaller burst size is subject to a minimum value that enables the system to achieve the configured rates.

In both configurations, the scheduler node can burst beyond its shaping rate for a brief interval. The burst of traffic beyond the shaping rate is more noticeable with the larger burst size than the smaller burst size.

Guidelines for Configuring the Burst Size

Typically, the default burst-size (100 ms) for both scheduler nodes and queues on MIC and MPC interfaces is adequate for most networks. However, if you have intermediate equipment in your network that has very limited buffering and is intolerant of bursts of traffic, you might want to configure a lower value for the burst size.

Use caution when selecting a different burst size for your network. A burst size that is too high can overwhelm downstream networking equipment, causing dropped packets and inefficient network operation. Similarly, a burst size that is too low can prevent the network from achieving your configured rate.

When configuring a burst size, keep the following considerations in mind:

- The system uses an algorithm to determine the actual burst size that is implemented for a node or queue. For example, to reach a shaping rate of 8 Mbps, you must allocate 1MB of rate credits every second. A shaping rate of 8 Mbps with a burst size of 500,000 bytes of rate-credit per seconds enables the system to transmit at most 500,000 bytes, or 4 Mbps. The system cannot implement a burst size that prevents the rate from being achieved.

  For more information, see "How the System Calculates the Burst Size" on page 157.

- There are minimum and maximum burst sizes for each platform, and different nodes and queue types have different scaling factors. For example, the system ensures the burst cannot be set lower than 1 Mbps for a shaping rate of 8 Mbps. To smoothly shape traffic, rate credits are sent much faster than once per second. The interval at which rate credits are sent varies depending on the platform, the type of rate, and the scheduler level.

- When you have configured adjustments for the shaping rate (either by percentage or through an application such as ANCP or Multicast OIF), the system bases the default and minimum burst-size calculations on the adjusted shaping rate.

- When you have configured cell shaping mode to account for ATM cell tax, the system bases the default and minimum burst-size calculations on the post-tax shaping rate.

- The guaranteed rate and shaping rate share the value specified for the burst size. If the guaranteed rate has a burst size specified, that burst size is used for the shaping rate; if the shaping rate has a burst size specified, that bursts size is used for the guaranteed rate. If you have specified a burst size for both rates, the system uses the lesser of the two values.

- The burst size configured for the guaranteed rate cannot exceed the burst-size configured for the shaping rate. Starting in Junos OS Release 15.1, the CLI no longer generates a commit error when the

guaranteed-rate burst size is statically configured to be more than the shaping-rate burst size. This behavior changed with the advent of enhanced subscriber management. The system logs an error when the guaranteed-burst rate is higher, whether it is configured statically, dynamically with predefined variables, or by means of a change of authorization request.

- If you have not configured a guaranteed rate, logical interfaces and interface sets receive a default guaranteed rate from the port speed. Queues receive a default guaranteed rate from the parent *logical interface* or interface set.
- Burst-size is not supported with `per-priority-shaping`.

How the System Calculates the Burst Size

When calculating the burst size, the system uses an exponent of a power of two. For example:

```
Shaping-rate in bps * 100 ms / (8 bits/byte * 1000 ms/s) = 1,875,000 bytes
```

The system then rounds this value up. For example, the system uses the following calculation to determine the burst size for a scheduler node with a shaping rate of 150 Mbps:

```
Max (Shaping rate, Guaranteed rate) bps * 100 ms / (8 bits/byte * 1000 ms/s) = 1,875,000 bytes

Rounded up to the next higher power of two = 2,097,150 (which is 2**21, or 0x200000)
```

The system assigns a single burst size to each of the following rate pairs:

- Shaping rate and guaranteed rate
- Guaranteed high (GH) and guaranteed medium (GM)
- Excess high (EH) and excess low (EL)
- Guaranteed low (GL)

To calculate the burst size for each pair, the system:

- Uses the configured burst-size if only one of the pair is configured.
- Uses the lesser of the two burst sizes if both values are configured.
- Uses the next lower power of two.
- To calculate the minimum burst size, the system uses the greater of the two rates.

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, the CLI no longer generates a commit error when the guaranteed-rate burst size is statically configured to be more than the shaping-rate burst size.

RELATED DOCUMENTATION

- Per-Priority Shaping on MIC and MPC Interfaces Overview
- Managing Excess Bandwidth Distribution on Static Interfaces on MICs and MPCs

Managing Excess Bandwidth Distribution for Dynamic CoS on MIC and MPC Interfaces

Service providers often used tiered services that must utilize excess bandwidth as traffic patterns vary. By default, excess bandwidth between a configured guaranteed rate and shaping rate is shared equally among all queues with the same excess priority value, which might not be optimal for all subscribers to a service.

This feature is supported for MIC and MPC interfaces on MX Series routers.

To configure parameters to manage excess bandwidth for subscriber interfaces:

1. Configure the parameters for the interface.
   a. Configure the guaranteed and shaping rates.
      i. Configure the guaranteed rate:

```
[edit dynamic-profiles profile-name class-of-service traffic-control-profiles profile-name]
user@host# set guaranteed-rate(rate | $junos-cos-guaranteed-rate) <burst-size (bytes | $junos-cos-guaranteed-rate-burst)>
```

      ii. Configure the shaping rate:

```
[edit dynamic-profiles profile-name class-of-service traffic-control-profiles profile-name]
```

```
user@host# set shaping-rate (rate | $junos-cos-shaping-rate) <burst-size (bytes |
$junos-cos-shaping-rate-burst)>
```

TIP: On MPC/MIC interfaces, the guaranteed rate and the shaping rate share the value specified for the burst size. If the guaranteed rate has a burst size specified, it is used for the shaping rate; if the shaping rate has a burst size specified, it is used for the guaranteed rate. If you have specified a burst for both rates, the system uses the lesser of the two values.

b. Configure a rate for excess bandwidth.

You can configure an excess rate for all priorities of traffic:

```
[edit dynamic-profiles profile-name class-of-service traffic-control-profiles profile-name]
user@host# set excess-rate (percent percentage | $junos-cos-excess-rate) | proportion
value )
```

Optionally, you can configure an excess rate specifically for high- and low-priority traffic. When you configure the `excess-rate` statement for an interface, you cannot also configure the `excess-rate-low` and `excess-rate-high` statements.

```
[edit dynamic-profiles profile-name class-of-service traffic-control-profiles profile-name]
user@host# set excess-rate-high(percent percentage | $junos-cos-excess-rate-high) |
proportion value )
user@host# set excess-rate-low (percent percentage | $junos-cos-excess-rate-low) |
proportion value )
```

BEST PRACTICE: We recommend that you configure either a percentage or a proportion of the excess bandwidth for all schedulers with the same parent in the hierarchy. For example, if you configure interface 1.1 with twenty percent of the excess bandwidth, configure interface 1.2 with eighty percent of the excess bandwidth.

2. (Optional) Configure parameters for the queue.

a. Configure the shaping rate.

```
[edit dynamic-profiles profile-name class-of-service scheduler scheduler-name]
user@host#set shaping-rate (rate | $junos-cos-scheduler-shaping-rate) <burst-size bytes>
```

b. Configure the excess rate.

```
[edit dynamic-profiles profile-name class-of-service scheduler scheduler-name]
user@host#set excess-rate (percent percentage | percent $junos-cos-scheduler-excess-rate)
```

c. (Optional) Configure the priority of excess bandwidth for the queue.

```
[edit dynamic-profiles profile-name class-of-service scheduler scheduler-name]
user@host#set excess-priority (low | high | $junos-cos-scheduler-excess-priority | none)
```

TIP: For queues, you cannot configure the excess rate or excess priority in these cases:

- When the `transmit-rate exact` statement is configured. In this case, the shaping rate is equal to the transmit rate and the queue does not operate in the excess region.
- When the scheduling priority is configured as `strict-high`. In this case, the queue gets all available bandwidth and never operates in the excess region.

By default, when traffic exceeds the shaping or guaranteed rates, the system demotes traffic configured with high or medium priority. To disable priority demotion, specify the `none` option. You cannot configure this option for queues configured with `transmit-rate` expressed as a percent and when the parent's guaranteed rate is set to zero.

RELATED DOCUMENTATION

Guidelines for Configuring Dynamic CoS for Subscriber Access | 44

CHAPTER 11

Applying CoS Using Parameters Received from RADIUS

IN THIS CHAPTER

- Subscriber Interfaces That Provide Initial CoS Parameters Dynamically Obtained from RADIUS | **161**
- Changing CoS Services Overview | **166**
- CoS Traffic Shaping Attributes for Dynamic Interface Sets and Member Subscriber Sessions Overview | **170**
- Guidelines for Configuring CoS Traffic Shaping Attributes for Dynamic Interface Sets and Member Subscriber Sessions | **172**
- Configuring Initial CoS Parameters Dynamically Obtained from RADIUS | **173**
- Configuring Static Default Values for Traffic Scheduling and Shaping | **174**
- Applying CoS Traffic-Shaping Attributes to Dynamic Interface Sets and Member Subscriber Sessions | **176**
- CoS Traffic Shaping Predefined Variables for Dynamic Interface Sets | **179**
- Example: Configuring Initial CoS Parameters Dynamically Obtained from RADIUS | **185**

Subscriber Interfaces That Provide Initial CoS Parameters Dynamically Obtained from RADIUS

IN THIS SECTION

- Dynamic Configuration of Initial CoS in Access Profiles | **162**
- Predefined Variables for Dynamic Configuration of Initial Traffic Shaping | **162**
- Predefined Variables for Dynamic Configuration of Initial Scheduling and Queuing | **163**

You can configure interface-specific CoS parameters that the router obtains when subscribers log in at appropriately configured static or dynamic subscriber interfaces.

To configure a dynamic profile to provide initial CoS Services, make sure you understand the following concepts:

Dynamic Configuration of Initial CoS in Access Profiles

When a router interface receives a join message from a DHCP subscriber, Junos applies the values you configured in the dynamic profile associated with that router interface. A dynamic profile that activates through its association with a subscriber interface is known as an *access dynamic profile*. You can associate a dynamic profile with a subscriber interface on the router by including statements at the `[edit dynamic-profiles profile-name class-of-service interfaces]` hierarchy level.

Junos supports predefined variables for obtaining CoS parameters from the RADIUS authentication server. When a client authenticates over a router interface associated with the access dynamic profile, the router replaces the predefined variables with interface-specific values obtained from the RADIUS server.

NOTE: To associate dynamically configured initial CoS features with a subscriber interface, reference *Junos OS predefined variables*—and not *user-defined variables*—in an *access* dynamic profile for that interface.

Predefined Variables for Dynamic Configuration of Initial Traffic Shaping

You can configure an access dynamic profile that provides initial traffic-shaping parameters when a subscriber logs in. The Junos OS obtains this information from the RADIUS server when a subscriber authenticates over the static or dynamic subscriber interface to which the access dynamic profile is attached.

If you define the Juniper Networks authentication and authorization VSA for CoS traffic-shaping parameter values (attribute number 26–108) on the RADIUS authentication server, the RADIUS server includes the values in RADIUS Access-Accept messages it sends to the router when a subscriber successfully authenticates over the interface.

To provide an initial scheduler map name and traffic shaping parameters obtained from the RADIUS authentication server when a subscriber logs in, reference the Junos predefined variables for CoS listed in Table 18 on page 163 in an access dynamic profile associated with the subscriber interface.

Table 18: CoS Predefined Variables for Scheduler Map and Traffic Shaping

Variable	Description
$junos-cos-scheduler-map	Scheduler-map name to be dynamically configured in a traffic-control profile in the access dynamic profile when a subscriber logs in. NOTE: You can define the scheduler map referenced by the `scheduler-map` statement dynamically (at the `[edit dynamic-profiles profile-name class-of-service scheduler-maps]` hierarchy level) or statically (at the `[edit class-of-service scheduler-maps]` hierarchy level).
$junos-cos-shaping-rate	Shaping rate to be dynamically configured in a traffic-control profile in the access dynamic profile when a subscriber logs in. You can configure a RADIUS authentication server to include this information in the Accept-Accept message when a subscriber successfully authenticates over the static or dynamic subscriber interface to which the access dynamic profile is attached.
$junos-cos-guaranteed-rate	Guaranteed rate to be dynamically configured in a traffic-control profile in the access dynamic profile when a subscriber logs in. You can configure a RADIUS authentication server to include this information in the Accept-Accept message when a subscriber successfully authenticates over the static or dynamic subscriber interface to which the access dynamic profile is attached.
$junos-cos-delay-buffer-rate	Delay-buffer rate to be dynamically configured in a traffic-control profile in the access dynamic profile when a subscriber logs in. You can configure a RADIUS authentication server to include this information in the Accept-Accept message when a subscriber successfully authenticates over the static or dynamic subscriber interface to which the access dynamic profile is attached.

Predefined Variables for Dynamic Configuration of Initial Scheduling and Queuing

You can configure an access dynamic profile that provides initial traffic-shaping parameters when a subscriber logs in. Junos obtains this information from the RADIUS server when a subscriber authenticates over the static or dynamic subscriber interface to which the access dynamic profile is attached.

If you define the Juniper Networks authentication and authorization VSA for CoS scheduling and queuing parameter values (attribute number 26–146) on the RADIUS authentication server, the RADIUS server includes the values in RADIUS Access-Accept messages it sends to the router when a subscriber successfully authenticates over the interface.

To provide an initial scheduler name and scheduler and queuing parameters obtained from the RADIUS authentication server when a subscriber logs in, reference the Junos OS predefined variables listed in Table 19 on page 164 in an access dynamic profile associated with the subscriber interface.

Table 19: CoS Predefined Variables for Scheduling and Queuing

Variable	Description
$junos-cos-scheduler	Name of a scheduler to be dynamically configured in the access dynamic profile. You can configure a RADIUS authentication server to include this information in the Accept-Accept message when a subscriber successfully authenticates over the static or dynamic subscriber interface to which the access dynamic profile is attached.
$junos-cos-scheduler-transmit-rate	Transmit rate to be dynamically configured for the scheduler in the access dynamic profile. You can configure a RADIUS authentication server to include this information in the Accept-Accept message when a subscriber successfully authenticates over the static or dynamic subscriber interface to which the access dynamic profile is attached.
$junos-cos-scheduler-bs	Buffer size, as a percentage of total buffer, to be dynamically configured for the scheduler in the access dynamic profile. You can configure a RADIUS authentication server to include this information in the Accept-Accept message when a subscriber successfully authenticates over the static or dynamic subscriber interface to which the access dynamic profile is attached.
$junos-cos-scheduler-pri	Packet-scheduling priority value to be dynamically configured for the scheduler in the access dynamic profile. You can configure a RADIUS authentication server to include this information in the Accept-Accept message when a subscriber successfully authenticates over the static or dynamic subscriber interface to which the access dynamic profile is attached.
$junos-cos-scheduler-dropfile-low	Name of the drop profile for RED for loss-priority level `low` to be dynamically configured for the scheduler in the access dynamic profile. You can configure a RADIUS authentication server to include this information in the Accept-Accept message when a subscriber successfully authenticates over the static or dynamic subscriber interface to which the access dynamic profile is attached. `NOTE`: The drop profile must be configured statically (at the `[edit class-of-service schedulers` *`scheduler-name`* `drop-profiles]` hierarchy level) for loss-priority `low`.

Table 19: CoS Predefined Variables for Scheduling and Queuing *(Continued)*

Variable	Description
$junos-cos-scheduler-dropfile-medium-low	Name of the drop profile for RED for loss-priority level `medium-low` to be dynamically configured for the scheduler in the access dynamic profile. The Junos OS obtains this information from the RADIUS server when a subscriber authenticates over the static or dynamic subscriber interface to which the access dynamic profile is attached. `NOTE:` The drop profile must be configured statically (at the `[edit class-of-service schedulers` *`scheduler-name`* `drop-profiles]` hierarchy level).
$junos-cos-scheduler-dropfile-medium-high	Name of the drop profile for RED for loss-priority level `medium-high` to be dynamically configured for the scheduler in the access dynamic profile. You can configure a RADIUS authentication server to include this information in the Accept-Accept message when a subscriber successfully authenticates over the static or dynamic subscriber interface to which the access dynamic profile is attached. `NOTE:` The drop profile must be configured statically (at the `[edit class-of-service schedulers` *`scheduler-name`* `drop-profiles]` hierarchy level).
$junos-cos-scheduler-dropfile-high	Name of the drop profile for RED for loss-priority level `high` to be dynamically configured for the scheduler in the access dynamic profile. You can configure a RADIUS authentication server to include this information in the Accept-Accept message when a subscriber successfully authenticates over the static or dynamic subscriber interface to which the access dynamic profile is attached. `NOTE:` The drop profile must be configured statically (at the `[edit class-of-service schedulers` *`scheduler-name`* `drop-profiles]` hierarchy level).
$junos-cos-scheduler-dropfile-any	Name of the drop profile for RED for loss-priority level `any` to be dynamically configured for the scheduler in the access dynamic profile. You can configure a RADIUS authentication server to include this information in the Accept-Accept message when a subscriber successfully authenticates over the static or dynamic subscriber interface to which the access dynamic profile is attached. `NOTE:` The drop profile must be configured statically (at the `[edit class-of-service schedulers` *`scheduler-name`* `drop-profiles]` hierarchy level).

RELATED DOCUMENTATION

Subscriber Activation and Service Management in an Access Network

Dynamic Profiles Overview

Dynamic Variables Overview

Junos OS Predefined Variables

Configuring Initial CoS Parameters Dynamically Obtained from RADIUS | 173

Example: Configuring Initial CoS Parameters Dynamically Obtained from RADIUS | 185

Changing CoS Services Overview

IN THIS SECTION

- Types of CoS Variables Used in a Service Profile | 167
- Static and Dynamic CoS Configurations | 167
- Scenarios for Static and Dynamic Configuration of CoS Parameters | 168

This topic describes how to provide CoS when subscribers dynamically upgrade or downgrade services in an access environment.

You can configure your network with a *dynamic client profile* that provides all subscribers with default CoS parameters when they log in. For example, all subscribers can receive a basic data service. By configuring the client profile with Junos OS predefined variables for RADIUS-provided CoS parameters, you also enable the service to be activated for those subscribers at login.

NOTE: The dynamic client profile is also referred to as a dynamic client access profile, or sometimes just access profile for brevity. Do not confuse this profile, configured at the `[edit dynamic-profiles profile-name]` hierarchy level, with the access profile configured at the `[edit access profile profile-name]` hierarchy level. These static access profiles are used to configure authentication, accounting, and authorization parameters for subscriber access, some session attributes, and client-specific properties for L2TP and PPP sessions. Access profiles are applied at various configuration levels with the `access-profile` statement.

To enable subscribers to activate a service or upgrade to different services through RADIUS change-of-authorization (CoA) messages after login, configure a *dynamic service profile* that includes user-defined variables.

Types of CoS Variables Used in a Service Profile

You can configure variables for the following CoS parameters in a service profile:

- Shaping rate
- Delay buffer rate
- Guaranteed rate
- Scheduler map

For each CoS parameter, you must associate a RADIUS vendor ID. For each vendor ID, you must assign an attribute number and a tag. The tag is used to differentiate between values for different CoS variables when you specify the same attribute number for those variables. These values are matched with the values supplied by RADIUS during subscriber authentication. All of the values in the dynamic profile must be defined in RADIUS or none of the values are passed.

Optionally, you can configure default values for each parameter. Configuring default values is beneficial if you do not configure RADIUS to enable service changes. During service changes, RADIUS takes precedence over the default value that is configured.

Static and Dynamic CoS Configurations

Depending on how you configure CoS parameters in the access and service profiles, certain CoS parameters are replaced or merged when subscribers change or activate new services.

Static configuration is when you configure the scheduler map and schedulers in the static `[edit class-of-service]` hierarchy and reference the scheduler map in the dynamic profile. Dynamic configuration is when you configure the scheduler map and schedulers within the dynamic profile.

The CoS configuration also depends on whether you have enabled multiple subscribers on the same *logical interface* using the `aggregate-clients` statements in the dynamic profile referenced by DHCP. When you specify the `aggregate-clients replace` statement, the scheduler map names are replaced. In both cases, if the length of the scheduler map name exceeds 128 characters, subscribers cannot log in. When you specify the `aggregate-clients merge` statement, the scheduler map names specified in the dynamic profile are appended.

BEST PRACTICE: To improve CoS performance in IPv4, IPv6, and dual-stack networks, we recommend that you use the `aggregate-clients replace` statement rather than the `aggregate-clients merge` statement.

Scenarios for Static and Dynamic Configuration of CoS Parameters

Table 20 on page 168 lists the scenarios for static and dynamic configuration of CoS parameters in access profiles and service profiles at subscriber login. The table also lists the behavior for each configuration for service activation and service modification using RADIUS CoA messages.

Table 20: CoS Services and Variables

Scenario	Static CoS Configuration (Single Subscriber)	Dynamic CoS Configuration (Single Subscriber)	Dynamic CoS Configuration (Multiple Subscribers Enabled on a Logical Interface with the `aggregate-clients merge` Statement)	Dynamic CoS Configuration (Multiple Subscribers Enabled on a Logical Interface with the `aggregate-clients replace` Statement)
Subscriber login	• Configure RADIUS values or default values for all parameters in access profile • Configure scheduler map in `edit class-of-service` hierarchy and reference in access profile	• Configure RADIUS values or default values for all parameters in access profile • Configure scheduler map and schedulers in access profile	• Configure RADIUS values or default values for all parameters in access profile • Configure scheduler map and schedulers in access profile	• Configure RADIUS values or default values for all parameters in access profile • Configure scheduler map and schedulers in access profile

Table 20: CoS Services and Variables *(Continued)*

Scenario	Static CoS Configuration (Single Subscriber)	Dynamic CoS Configuration (Single Subscriber)	Dynamic CoS Configuration (Multiple Subscribers Enabled on a Logical Interface with the `aggregate-clients merge` Statement)	Dynamic CoS Configuration (Multiple Subscribers Enabled on a Logical Interface with the `aggregate-clients replace` Statement)
RADIUS CoA for service or variable change	Replaces the following parameters: • Delay buffer rate • Guaranteed rate • Scheduler map • Shaping rate	Replaces the following parameters: • Delay buffer rate • Guaranteed rate • Shaping rate • Scheduler map	Combines the values of the following parameters to their maximum scalar value: • Delay buffer rate • Guaranteed rate • Shaping rate Appends the scheduler map parameter	Replaces the following parameters: • Delay buffer rate • Guaranteed rate • Shaping rate • Scheduler map
RADIUS CoA for service activation	Does not merge queues `NOTE`: In this case, use a similar configuration to the access profile, including the same name for the traffic-control-profile. During service activation, this configuration replaces the original configuration in the access profile.	Merge queues if the queue specified in the service profile is not already in use for the subscriber `NOTE`: Do not instantiate a CoA request using a service dynamic profile that is already in use on the same logical interface.	Merge queues if the queue specified in the service profile is not already in use for the subscriber `NOTE`: Do not instantiate a CoA request using a service dynamic profile that is already in use on the same logical interface.	Merge queues if the queue specified in the service profile is not already in use for the subscriber `NOTE`: Do not instantiate a CoA request using a service dynamic profile that is already in use on the same logical interface.

RELATED DOCUMENTATION

- Configuring Static Hierarchical Scheduling in a Dynamic Profile
- *Dynamic Profile Attachment to DHCP Subscriber Interfaces Overview*
- *RADIUS Attributes and Juniper Networks VSAs Supported by the AAA Service Framework*
- Guidelines for Configuring Dynamic CoS for Subscriber Access | **44**

CoS Traffic Shaping Attributes for Dynamic Interface Sets and Member Subscriber Sessions Overview

IN THIS SECTION

- Supported Network Configurations | **170**
- Traffic-Control Profiles in Subscriber Interface Dynamic Profiles | **171**
- CoS Traffic Shaping Predefined Variables for Dynamic Interface Sets and Member Subscriber Sessions | **171**

To control bandwidth at a household level in a subscriber access network, you can apply RADIUS dynamic *class of service* (CoS) traffic-shaping attributes to a dynamic interface set and its member subscriber sessions when the subscriber sessions are authenticated. (The dynamic interface set itself does not go through the authentication process.)

A *household* is represented by either a dynamic interface set or a dynamic agent-circuit-identifier (ACI) interface set from which the subscriber sessions originate. For this feature, dynamic interface sets and dynamic ACI interface sets are mapped to Level 2 of the Junos OS CoS scheduler hierarchy, which enables you to use CoS traffic-shaping to shape the bandwidth at the household (interface set) level.

The *subscriber sessions*, also referred to as *subscriber interfaces* or *client sessions*, can be dynamic VLAN, PPPoE, or IP demultiplexing (IP demux) subscriber interfaces. The subscriber interfaces are mapped to Level 3 of the Junos OS CoS scheduler hierarchy.

Supported Network Configurations

Applying RADIUS dynamic CoS traffic-shaping attributes to a dynamic interface set and its member subscriber sessions is supported for the following network configurations:

- Dynamic IP demux subscriber interfaces (for DHCP subscribers) over either a dynamic interface set or a dynamic ACI interface set

- Dynamic PPPoE subscriber interfaces over either a dynamic interface set or a dynamic ACI interface set

Traffic-Control Profiles in Subscriber Interface Dynamic Profiles

To apply dynamic CoS traffic-shaping attributes to a dynamic interface set and its member subscriber sessions, you must define and attach the traffic-control profiles for *both* the dynamic interface set and the dynamic subscriber sessions within the dynamic profile for the subscriber interface.

At the `[edit dynamic-profiles profile-name class-of-service traffic-control-profiles]` hierarchy level in the dynamic profile, configure both of the following:

- Traffic-control profile for the dynamic VLAN, PPPoE, or IP demux subscriber interfaces
- Traffic-control profile for the dynamic interface set or dynamic ACI interface set to which the subscriber interfaces belong

RADIUS tag values for the Junos OS CoS traffic shaping predefined variables used in both traffic-control profiles must be in the 100s range, as described in "CoS Traffic Shaping Predefined Variables for Dynamic Interface Sets" on page 179.

At the `[edit dynamic-profiles profile-name interfaces]` hierarchy level in the dynamic profile, use the `output-traffic-control-profile` statement to apply the traffic-control profiles to the dynamic subscriber interface and the dynamic interface set or dynamic ACI interface set.

CoS Traffic Shaping Predefined Variables for Dynamic Interface Sets and Member Subscriber Sessions

The set of `$junos-cos-parameter` predefined dynamic variables has been duplicated and assigned a RADIUS tag value in the 100s range for use with this feature. The RADIUS tag value is the only difference between the existing CoS traffic-shaping predefined dynamic variables and the predefined dynamic variables that you must use with this feature.

Both RADIUS instances of the `$junos-cos-parameter` predefined dynamic variables are available, but you must use the dynamic variables with tag values in the 100s range to apply CoS traffic-shaping attributes to both the dynamic interface set and member subscriber sessions in a subscriber interface dynamic profile.

For example, the existing `$junos-cos-shaping-rate` predefined variable is assigned RADIUS vendor ID 4874, attribute number 108, and tag value 2. To apply CoS traffic-shaping attributes to the dynamic interface set and its member subscriber sessions, you must instead use the `$junos-cos-shaping-rate` predefined variable that is assigned RADIUS vendor ID 4874, attribute number 108, and tag value 102.

NOTE: Do not configure a combination of `$junos-cos-parameter` predefined dynamic variables with RADIUS tag values in the 100s range and `$junos-cos-parameter` predefined dynamic variables with tag values not in the 100s range in the same traffic-control profile. If you do so, the subscriber authentication process fails.

RELATED DOCUMENTATION

Guidelines for Configuring CoS Traffic Shaping Attributes for Dynamic Interface Sets and Member Subscriber Sessions | 172

Applying CoS Traffic-Shaping Attributes to Dynamic Interface Sets and Member Subscriber Sessions | 176

CoS Traffic Shaping Predefined Variables for Dynamic Interface Sets | 179

Guidelines for Configuring CoS Traffic Shaping Attributes for Dynamic Interface Sets and Member Subscriber Sessions

Observe the following guidelines when you apply dynamic CoS traffic-shaping attributes to a dynamic interface set or a dynamic ACI interface set and its member subscriber sessions. For complete information about the Junos OS CoS traffic-shaping predefined dynamic variables and RADIUS tag values used with this feature, see "CoS Traffic Shaping Predefined Variables for Dynamic Interface Sets" on page 179.

- This feature is supported only for dynamically configured and instantiated subscriber interfaces.
- Do not configure a combination of `$junos-cos-parameter` predefined dynamic variables with RADIUS tag values in the 100s range and `$junos-cos-parameter` predefined dynamic variables with tag values not in the 100s range in the same traffic-control profile (TCP). If you do so, the subscriber authentication process fails.
- Use the `$junos-cos-adjust-minimum` predefined variable (tag 109) only in TCPs for dynamic subscriber interfaces. Using this variable in a TCP for a dynamic interface set or dynamic ACI interface set has no effect.
- Do not configure the `$junos-cos-excess-rate-high` predefined variable (tag 110) and the `$junos-cos-excess-rate` predefined variable (tag 105) in the same TCP.
- Do not configure the `$junos-cos-excess-rate-low` predefined variable (tag 111) and the `$junos-cos-excess-rate` predefined variable (tag 105) in the same TCP.

- Do not configure the `$junos-cos-byte-adjust-frame` predefined variable (tag 114) and the `$junos-cos-byte-adjust` predefined variable (tag 108) in the same TCP.
- Do not configure the `$junos-cos-byte-adjust-cell` predefined variable (tag 115) and the `$junos-cos-byte-adjust` predefined variable (tag 108) in the same TCP.
- Use the per-priority `$junos-cos-shaping-rate-`*`parameter`* predefined variables (tags 116 through 125) only in TCPs for dynamic interface sets or dynamic ACI interface sets. Using these variables in TCPs for a dynamic logical subscriber interface causes the subscriber session to fail.

RELATED DOCUMENTATION

Applying CoS Traffic-Shaping Attributes to Dynamic Interface Sets and Member Subscriber Sessions | **176**

CoS Traffic Shaping Predefined Variables for Dynamic Interface Sets | **179**

CoS Traffic Shaping Attributes for Dynamic Interface Sets and Member Subscriber Sessions Overview | **170**

Configuring Initial CoS Parameters Dynamically Obtained from RADIUS

You can configure a subscriber interface so that subscribers receive initial CoS parameters that the router obtains from the RADIUS authentication server when subscribers log in using that logical interface on the router.

1. Configure external RADIUS server VSAs with values that you expect subscribers to log in with.
   - To configure a RADIUS authentication server to include CoS traffic-shaping parameters in authentication grants on certain subscriber interfaces, configure Juniper Networks VSA 26–108.
   - To configure a RADIUS authentication server to include CoS scheduling and queuing parameters in authentication grants a certain subscriber interfaces, configure Juniper Networks VSA 28–146.

   See Configuring Router or Switch Interaction with RADIUS Servers and *RADIUS Servers and Parameters for Subscriber Access*.
2. Configure a subscriber interface that supports hierarchical CoS.
3. Associate a traffic-control profile with the interface.

   See "Applying Traffic Shaping and Scheduling to a Subscriber Interface in a Dynamic Profile" on page 228.
4. Configuring initial traffic-shaping parameters to be obtained from RADIUS.

   See "Configuring Dynamic Traffic Shaping and Scheduling Parameters in a Dynamic Profile" on page 54.

5. Configure forwarding classes and scheduler maps statically.

   See Configuring a Custom Forwarding Class for Each Queue and Configuring Scheduler Maps.

6. Configure a scheduler to specify initial scheduling and queuing parameters to be dynamically obtained from RADIUS when a subscriber logs in.

   See "Configuring Dynamic Schedulers with Variables in a Dynamic Profile" on page 60.

RELATED DOCUMENTATION

Subscriber Interfaces That Provide Initial CoS Parameters Dynamically Obtained from RADIUS | 161

Example: Configuring Initial CoS Parameters Dynamically Obtained from RADIUS | 185

Guidelines for Configuring Dynamic CoS for Subscriber Access | 44

Subscriber Activation and Service Management in an Access Network

Juniper Networks VSAs Supported by the AAA Service Framework

Dynamic Profiles Overview

Dynamic Variables Overview

Junos OS Predefined Variables

Configuring Static Default Values for Traffic Scheduling and Shaping

To provide subscribers with default values for CoS parameters, configure user-defined variables for CoS parameters and assign static default values to the variables. If you have configured values to be supplied by a RADIUS CoA, subscribers receive the default value when deactivating a service.

To configure user-defined variables with default values for CoS in a dynamic profile:

1. Specify that you want to configure variables in the dynamic profile.

```
[edit dynamic-profiles residential-silver variables]
```

2. Configure a default value for the shaping rate.

```
[edit dynamic-profiles residential-silver variables]
user@host# set srate default-value 5m
```

3. Configure a default value for the guaranteed rate.

```
[edit dynamic-profiles residential-silver variables]
user@host# set grate default-value 5m
```

4. Configure a default value for the delay buffer rate.

```
[edit dynamic-profiles residential-silver variables]
user@host# set dbrate default-value 10m
```

5. Configure a default value for the scheduler map.

```
[edit dynamic-profiles residential-silver variables]
user@host# set smap default-value triple-play
```

6. Configure the variables for the CoS parameters in the traffic-control profile.

   Either the shaping rate or the guaranteed rate is required in the traffic-control profile.

   a. Access the traffic-control profile in the dynamic profile.

```
user@host# edit dynamic-profiles residential-silver class-of-service traffic-control-
profiles tcp1
```

   b. Configure the scheduler map variable.

```
[edit dynamic-profiles residential-silver class-of-service traffic-control-profiles tcp1]
user@host# set scheduler-map "$smap"
```

   c. Configure the shaping rate variable.

```
[edit dynamic-profiles residential-silver class-of-service traffic-control-profiles tcp1]
user@host# set shaping-rate "$srate"
```

   d. Configure the guaranteed rate variable.

```
[edit dynamic-profiles residential-silver class-of-service traffic-control-profiles tcp1]
user@host# set guaranteed-rate "$grate"
```

e. Configure the delay buffer rate variable.

```
[edit dynamic-profiles residential-silver class-of-service traffic-control-profiles tcp1]
user@host# set delay-buffer-rate "$dbrate"
```

RELATED DOCUMENTATION

Guidelines for Configuring Dynamic CoS for Subscriber Access | 44

Changing CoS Services Overview | 166

Applying CoS Traffic-Shaping Attributes to Dynamic Interface Sets and Member Subscriber Sessions

To control bandwidth at a household level in a subscriber access network, you can apply RADIUS dynamic class of service (CoS) traffic-shaping attributes to a dynamic interface set or agent-circuit-identifer (ACI) interface set and its member subscriber sessions when the member sessions are authenticated. The dynamic interface set or ACI interface set represents the *household* from which the subscriber sessions originate. The *subscriber sessions*, also referred to as *client sessions* or *subscriber interfaces*, can be dynamic VLAN, PPPoE, or IP demultiplexing (IP demux, for DHCP) subscriber interfaces.

To apply RADIUS dynamic CoS traffic-shaping attributes to both the dynamic interface set and its member subscriber sessions, you must configure two traffic-control profiles in the dynamic profile for the subscriber interface: one traffic-control profile for the “parent” dynamic interface set, and a second traffic-control profile for the dynamic subscriber interfaces. RADIUS tag values for the Junos OS CoS traffic shaping predefined variables used in both traffic-control profiles must be in the 100s range.

Before you begin:

- Create a dynamic profile that defines the VLAN, PPPoE, or IP demux logical subscriber interface.

  See the following topics:

  - *Configuring a Basic Dynamic Profile*
  - *Configuring a Dynamic Profile Used to Create Single-Tag VLANs*
  - *Configuring a Dynamic Profile Used to Create Stacked VLANs*
  - *Configuring Dynamic PPPoE Subscriber Interfaces*

To apply dynamic CoS traffic-shaping attributes to a dynamic ACI or non-ACI interface set and its member subscriber sessions in a dynamic profile for the subscriber interface:

1. Configure two traffic-control profiles at the `[edit dynamic-profiles *profile-name* class-of-service traffic-control profiles]` hierarchy level:
   - Traffic-control profile for the VLAN, PPPoE, or IP demux dynamic subscriber interfaces
   - Traffic-control profile for the dynamic interface set or dynamic ACI interface set to which the subscriber interfaces belong
2. In the traffic-control profiles configured for the dynamic interface set and the subscriber interfaces, reference Junos OS CoS traffic-shaping predefined variables with RADIUS tag values in the 100s range.

   See "CoS Traffic Shaping Predefined Variables for Dynamic Interface Sets" on page 179 for a complete list of the Junos OS predefined variables and RADIUS tag values that you must use in the traffic-control profiles for the dynamic subscriber interfaces and the dynamic interface set.
3. At the `[edit dynamic-profiles *profile-name* interfaces]` hierarchy level, use the `output-traffic-control-profile` statement to apply the traffic-control profiles to the dynamic subscriber interface and the dynamic interface set or dynamic ACI interface set.

Example: Dynamic PPPoE Subscriber Interface over Dynamic ACI Interface Set

The following example shows a dynamic profile named pppoe-subscriber that configures a dynamic PPPoE (`pp0`) subscriber interface over a dynamic ACI interface set.

The `traffic-control-profiles` stanza defines two traffic-control profiles: tcp-pppoe-session for the dynamic PPPoE subscriber interface, and tcp-parent-aci-set for the dynamic “parent” ACI interface set. The `$junos-cos-shaping-rate` predefined variable included in each of these traffic-control profiles is assigned RADIUS vendor ID 4874, attribute number 108, and tag value 102. The `$junos-cos-shaping-mode` variable is assigned RADIUS vendor ID 4874, attribute number 108, and tag value 107.

The `interfaces` stanza applies output traffic-control profile tcp-pppoe-session to the dynamic PPPoE (`pp0`) subscriber interface, and output traffic-control profile tcp-parent-aci-set to the dynamic ACI interface set.

```
[edit dynamic-profiles]
pppoe-subscriber {
    interfaces {
        interface-set "$junos-interface-set-name" {
            interface pp0 {
                unit "$junos-interface-unit";
            }
        }
        pp0 {
```

```
            unit "$junos-interface-unit" {
                ppp-options {
                    pap;
                }
                pppoe-options {
                    underlying-interface "$junos-underlying-interface";
                    server;
                }
                no-keepalives;
                family inet {
                    unnumbered-address lo0.0;
                }
            }
        }
    }
    class-of-service {
        traffic-control-profiles {
            tcp-pppoe-session {
                scheduler-map smap-1;
                shaping-rate $junos-cos-shaping-rate;
                overhead-accounting $junos-cos-shaping-mode frame-mode-bytes -4 cell-mode-bytes
12;
            }
            tcp-parent-aci-set {
                shaping-rate $junos-cos-shaping-rate;
                overhead-accounting $junos-cos-shaping-mode frame-mode-bytes -4 cell-mode-bytes
12;
            }
        }
        interfaces {
            pp0 {
                unit "$junos-interface-unit" {
                    output-traffic-control-profile tcp-pppoe-session;
                }
            }
            interface-set $junos-interface-set-name {
                output-traffic-control-profile tcp-parent-aci-set;
            }
        }
    }
    }
}
```

RELATED DOCUMENTATION

CoS Traffic Shaping Predefined Variables for Dynamic Interface Sets | 179

CoS Traffic Shaping Attributes for Dynamic Interface Sets and Member Subscriber Sessions Overview | 170

Guidelines for Configuring CoS Traffic Shaping Attributes for Dynamic Interface Sets and Member Subscriber Sessions | 172

CoS Traffic Shaping Predefined Variables for Dynamic Interface Sets

To control bandwidth at a household level in a subscriber access network, you can apply RADIUS CoS traffic-shaping attributes to a dynamic interface set and its member subscriber sessions when the member sessions are authenticated. The dynamic interface set, which represents the household level in a subscriber access network, can be either a dynamic agent-circuit-identifier (ACI) interface set or a non-ACI–based dynamic interface set. The subscriber sessions belonging to the interface set can be dynamic VLAN, DHCP, or PPPoE subscriber interfaces.

To apply RADIUS CoS traffic-shaping attributes to both the dynamic interface set and its member subscriber sessions, you must configure two traffic-control profiles in the dynamic profile for the subscriber interface: one traffic-control profile for the “parent” dynamic interface set, and a second traffic-control profile for the dynamic subscriber interfaces. RADIUS tag values for the Junos OS CoS traffic-shaping predefined variables used in these traffic-control-profiles must be in the 100s range, as described in Table 21 on page 180.

To accommodate this feature, the set of existing `$junos-cos-`*`parameter`* predefined dynamic variables for traffic shaping have been duplicated and assigned a tag value in the 100s range, as listed in Table 21 on page 180. The tag value is the only difference between the existing predefined dynamic variables and the predefined dynamic variables that you must use with this feature.

For example, the existing `$junos-cos-shaping-rate` predefined variable is assigned RADIUS vendor ID 4874, attribute number 108, and tag value 2. To apply RADIUS CoS traffic-shaping attributes to the dynamic interface set and its member subscriber sessions, you must instead use the `$junos-cos-shaping-rate` predefined variable that is assigned RADIUS vendor ID 4874, attribute number 108, and tag value 102.

Table 21 on page 180 describes the Junos OS predefined dynamic variables and RADIUS tag values that you can use in a dynamic profile to apply RADIUS CoS traffic-shaping attributes to the dynamic interface set and its member subscriber sessions. The table lists the predefined dynamic variables in ascending order by tag value.

NOTE: All of the predefined variables listed in Table 21 on page 180 use RADIUS vendor ID 4874 and RADIUS attribute value 108.

Table 21: Junos OS CoS Traffic Shaping Predefined Variables for Dynamic Interface Sets

Predefined Variable	RADIUS Tag Value	Description
$junos-cos-scheduler-map	101	Scheduler-map name configured in a traffic-control profile in a dynamic profile.
$junos-cos-shaping-rate	102	Shaping rate configured in a traffic-control profile in a dynamic profile. Represents the maximum bandwidth of a CoS scheduler node.
$junos-cos-guaranteed-rate	103	Guaranteed rate configured in a traffic-control profile in a dynamic profile. Represents the minimum bandwidth of a CoS scheduler node.
$junos-cos-delay-buffer-rate	104	Delay-buffer rate configured in a traffic-control profile in a dynamic profile.
$junos-cos-excess-rate	105	Excess rate configured in a traffic-control profile in a dynamic profile; scheduler weighting when operating in the excess region between the guranteed rate and the shaping rate. **NOTE**: Do not configure the $junos-cos-excess-rate variable when either the $junos-cos-excess-rate-high variable or the $junos-cos-excess-rate-low variable is configured.
$junos-cos-traffic-control-profile	106	Traffic-control profile configured in a dynamic profile for subscriber access.
$junos-cos-shaping-mode	107	Overhead-accounting mode configured in a traffic-control profile in a dynamic profile to shape downstream ATM traffic based on either frames (frame-mode) or cells (cell-mode).

Table 21: Junos OS CoS Traffic Shaping Predefined Variables for Dynamic Interface Sets *(Continued)*

Predefined Variable	RADIUS Tag Value	Description
$junos-cos-byte-adjust	108	Byte adjustment value for the cell or frame shaping mode configured in a traffic-control profile in a dynamic profile. **NOTE**: Do not configure the $junos-cos-byte-adjust variable when either the $junos-cos-byte-adjust-frame variable or the $junos-cos-byte-adjust-cell variable is configured.
$junos-cos-adjust-minimum	109	Minimum adjusted shaping rate configured in a traffic-control profile for a dynamic subscriber interface. Specifying this variable in a traffic-control profile for a dynamic interface set has no effect.
$junos-cos-excess-rate-high	110	Shaping rate configured for excess high-priority traffic in a traffic-control profile in a dynamic profile. **NOTE**: Do not configure the $junos-cos-excess-rate-high variable when the $junos-cos-excess-rate variable is configured.
$junos-cos-excess-rate-low	111	Shaping rate configured for excess low-priority traffic in a traffic-control profile in a dynamic profile. **NOTE**: Do not configure the $junos-cos-excess-rate-low variable when the $junos-cos-excess-rate variable is configured.
$junos-cos-shaping-rate-burst	112	Burst size for the shaping rate configured in a traffic-control profile in a dynamic profile.

Table 21: Junos OS CoS Traffic Shaping Predefined Variables for Dynamic Interface Sets *(Continued)*

Predefined Variable	RADIUS Tag Value	Description
$junos-cos-guaranteed-rate-burst	113	Burst size for the guaranteed rate configured in a traffic-control profile in a dynamic profile.
$junos-cos-byte-adjust-frame	114	Overhead bytes when downstream ATM traffic is in frame-mode. **NOTE**: Do not configure the $junos-cos-byte-adjust-frame variable when the $junos-cos-byte-adjust variable is configured.
$junos-cos-byte-adjust-cell	115	Overhead bytes when downstream ATM traffic is in cell-mode. **NOTE**: Do not configure the $junos-cos-byte-adjust-cell variable when the $junos-cos-byte-adjust variable is configured.
$junos-cos-shaping-rate-priority-high	116	Shaping rate configured for high-priority traffic in a traffic-control profile for a dynamic interface set or dynamic ACI interface set at a household level. Specifying this variable in a traffic-control profile for a dynamic subscriber interface is prohibited.
$junos-cos-shaping-rate-priority-high-burst	117	Shaping rate burst size configured for high-priority traffic in a traffic-control profile for a dynamic interface set or dynamic ACI interface set at a household level. Specifying this variable in a traffic-control profile for a dynamic subscriber interface is prohibited.

Table 21: Junos OS CoS Traffic Shaping Predefined Variables for Dynamic Interface Sets *(Continued)*

Predefined Variable	RADIUS Tag Value	Description
$junos-cos-shaping-rate-priority-medium	118	Shaping rate configured for medium-priority traffic in a traffic-control profile for a dynamic interface set or dynamic ACI interface set at a household level. Specifying this variable in a traffic-control profile for a dynamic subscriber interface is prohibited.
$junos-cos-shaping-rate-priority-medium-burst	119	Shaping rate burst size configured for medium-priority traffic in a traffic-control profile for a dynamic interface set or dynamic ACI interface set at a household level. Specifying this variable in a traffic-control profile for a dynamic subscriber interface is prohibited.
$junos-cos-shaping-rate-priority-low	120	Shaping rate configured for low-priority traffic in a traffic-control profile for a dynamic interface set or dynamic ACI interface set at a household level. Specifying this variable in a traffic-control profile for a dynamic subscriber interface is prohibited.
$junos-cos-shaping-rate-priority-low-burst	121	Shaping rate burst size configured for low-priority traffic in a traffic-control profile for a dynamic interface set or dynamic ACI interface set at a household level. Specifying this variable in a traffic-control profile for a dynamic subscriber interface is prohibited.

Table 21: Junos OS CoS Traffic Shaping Predefined Variables for Dynamic Interface Sets *(Continued)*

Predefined Variable	RADIUS Tag Value	Description
$junos-cos-shaping-rate-excess-high	122	Shaping rate configured for excess high-priority traffic in a traffic-control profile for a dynamic interface set or dynamic ACI interface set at a household level. Specifying this variable in a traffic-control profile for a dynamic subscriber interface is prohibited.
$junos-cos-shaping-rate-excess-high-burst	123	Shaping rate burst size configured for excess high-priority traffic in a traffic-control profile for a dynamic interface set or dynamic ACI interface set at a household level. Specifying this variable in a traffic-control profile for a dynamic subscriber interface is prohibited.
$junos-cos-shaping-rate-excess-low	124	Shaping rate configured for excess low-priority traffic in a traffic-control profile for a dynamic interface set or dynamic ACI interface set at a household level. Specifying this variable in a traffic-control profile for a dynamic subscriber interface is prohibited.
$junos-cos-shaping-rate-excess-low-burst	125	Shaping rate burst size configured for excess low-priority traffic in a traffic-control profile for a dynamic interface set or dynamic ACI interface set at a household level. Specifying this variable in a traffic-control profile for a dynamic subscriber interface is prohibited.

RELATED DOCUMENTATION

Applying CoS Traffic-Shaping Attributes to Dynamic Interface Sets and Member Subscriber Sessions | 176

CoS Traffic Shaping Attributes for Dynamic Interface Sets and Member Subscriber Sessions Overview | 170

Guidelines for Configuring CoS Traffic Shaping Attributes for Dynamic Interface Sets and Member Subscriber Sessions | 172

Junos OS Predefined Variables

Example: Configuring Initial CoS Parameters Dynamically Obtained from RADIUS

The following configuration is an example of a client dynamic profile in which initial CoS parameters are dynamically obtained from the RADIUS server when a subscriber authenticates over the interface to which the dynamic profile is applied.

For this example, assume that the RADIUS authentication server has been configured with traffic-shaping parameters (at Juniper Networks VSA 26-108) and CoS scheduling and queuing parameters (at Juniper Networks VSA 26–146).

The subscriber interface is a single-unit static gigabit Ethernet VLAN interface on an EQ DPC port:

```
[edit]
interfaces {
    ge-9/0/3 {
        hierarchical-scheduler;
        vlan-tagging;
        unit 100 {
            vlan-id 100;
            family inet {
                address 192.168.32.2/24;
            }
        }
    }
}
```

The client dynamic profile `residential_silver` attaches the traffic-control profile `tcp_1` to the subscriber interface that is defined in the dynamic profile using the `$junos-interface-ifd-name` predefined variable.

```
[edit]
dynamic-profiles {
    residential_silver {
```

```
        interfaces {
            “$junos-interface-ifd-name” {
                unit “$junos-underlying-interface-unit” {
                    family inet;
                }
            }
        }
        class-of-service {
            interfaces {
                “$junos-interface-ifd-name” {
                    unit “$junos-underlying-interface-unit” {
                        output-traffic-control-profile tcp_1;
                    }
                }
            }
        }
    }
}
```

The traffic-control profile `tcp_1`, references Junos OS predefined variables to obtain a scheduler-map name and traffic-shaping parameter values from RADIUS when a subscriber logs in. For this example, assume that the RADIUS server replaces the Junos OS predefined variable `$junos-cos-scheduler-map` scheduler-map name `business_smap_1`. The scheduler map `business_smap_1` is configured in the client dynamic profile:

```
[edit]
dynamic-profiles {
    residential_silver {
        class-of-service {
            traffic-control-profiles {
                tcp_1 {
                    scheduler-map “$junos-cos-scheduler-map”; # ’business_smap_1’
                    shaping-rate "$junos-cos-shaping-rate";
                    guaranteed-rate "$junos-cos-guaranteed-rate";
                    delay-buffer-rate "$junos-cos-delay-buffer-rate";
                }
            }
            scheduler-maps {
                business_smap_1 {
                    forwarding-class best-effort scheduler be_sched;
                    forwarding-class ef scheduler home_sched
                }
```

```
            }
        }
    }
}
```

A scheduler definition references Junos OS predefined variables to obtain scheduler configurations from RADIUS when a subscriber logs in. For this example, assume that the RADIUS server provides scheduler configurations for schedulers named be_sched and home_sched, which are included in the scheduler map business_smap_1:

```
[edit]
dynamic-profiles {
    residential_silver {
        class-of-service {
            schedulers {
                “$junos-cos-scheduler” { # ’be_sched’ and ’home_sched’
                    transmit-rate "$junos-cos-scheduler-tx";
                    buffer-size "$junos-cos-scheduler-bs";
                    priority "$junos-cos-scheduler-pri";
                    drop-profile-map loss-priority low protocol any drop-profile “$junos-cos-
scheduler-dropfile-low“;
                    drop-profile-map loss-priority medium-low protocol any drop-profile “$junos-
cos-scheduler-dropfile-medium-low“;
                    drop-profile-map loss-priority medium-high protocol any drop-profile “$junos-
cos-scheduler-dropfile-medium-high“;
                    drop-profile-map loss-priority high protocol any drop-profile “$junos-cos-
scheduler-dropfile-high“;
                }
            }
        }
    }
}
```

Static configurations for CoS consist of configurations for the forwarding classes used in the scheduler map business_smap_1 and configurations for drop-profile names provided by RADIUS for as part of the scheduler configurations provided (for be_sched and home_sched) when a subscriber logs in:

```
[edit]
    class-of-service {
        forwarding-classes {
            queue 0 best-effort;
```

```
            queue 1 ef;
        }
        drop-profiles {
            . . . configurations_for_drop_profile_names_provided_by_RADIUS . . .
        }
    }
}
```

RELATED DOCUMENTATION

Subscriber Activation and Service Management in an Access Network

Dynamic Profiles Overview

Dynamic Variables Overview

Junos OS Predefined Variables

Subscriber Interfaces That Provide Initial CoS Parameters Dynamically Obtained from RADIUS | **161**

Configuring Initial CoS Parameters Dynamically Obtained from RADIUS | **173**

CHAPTER 12

Modifying a Subscriber' s Shaping Characteristics After a Subscriber is Instantiated

IN THIS CHAPTER

- CoS Adjustment Control Profiles Overview | 189
- Configuring CoS Adjustment Control Profiles | 192
- Verifying the CoS Adjustment Control Profile Configuration | 194

CoS Adjustment Control Profiles Overview

IN THIS SECTION

- Applications and Associated Algorithms in Adjustment Control Profiles | 191
- CoS Shaping Rate Fallback Behavior | 192

CoS adjustment control profiles control which applications and algorithms can modify a subscriber's shaping characteristics after a subscriber is instantiated. Subscriber shaping characteristics are configured using the Junos OS CLI or by RADIUS messages. Adjustment control profiles enable subscriber shaping characteristics by to be adjusted by other applications like ANCP, PPPoE tags, DHCP tags, and RADIUS Change of Authorization (CoA) messages after a subscriber is instantiated. Adjustment control profiles are router-wide and apply to both static and dynamic interfaces.

Table 22 on page 190 describes the applications and their associated default algorithms that can be configured to perform rate adjustments after the subscriber is instantiated.

Table 22: Adjustment Control Profile Applications and Algorithms

Application	Default Priority	Default Algorithm	Description
RADIUS-CoA	1	Adjust-always	RADIUS CoA messages can update the subscriber's attributes (like shaping rate) after the subscriber is authenticated and QoS parameters (like shaping rate) are assigned.
ANCP	1	Adjust-always	The ANCP application can modify the existing shaping rate for both static and dynamic logical interfaces, and static interface sets. By default, ANCP can override all other applications. The shaping rate must be specified in order to override it.
DHCP	2	Adjust-less	The DHCP application can include DSL Forum VSA attributes in its discovery messages, DHCPDISCOVER for DHCPv4 and SOLICIT for DHCPv6. The attributes can modify the Junos OS CLI-configured shaping-rate value, as well as the RADIUS-supplied shaping-rate value. By default, these values can be modified by subsequent RADIUS CoA messages and DHCP actions. The DSL Forum VSAs are conveyed in DHCP option 82, suboption 9 (Vendor-Specific Information suboption) for DHCPv4 and in Option 17 (Vendor-Specific Information option) for DHCPv6.
PPPoE-Tags	2	Adjust-less	The PPPoE IA tag access-rate-downstream can modify the Junos OS CLI-configured shaping-rate value, as well as the RADIUS-supplied shaping-rate value. By default, these values can be modified by subsequent RADIUS CoA messages and ANCP actions. These values are conveyed in PPPoE (PADI) discovery packets.

The lower the priority value, the higher the priority. For example priority 1 is higher than priority 2. By default, the application shaping rates compare as follows:

- ANCP has priority over all the other applications.
- RADIUS CoA has priority over DHCP tags or PPPoE IA tags.
- The DHCP tags or PPPoE IA tags have priority over the shaping rate configured in the traffic control profile.

Applications and Associated Algorithms in Adjustment Control Profiles

You must enable each application to perform rate adjustments. Rate adjustments are global and affect all static and dynamically instantiated subscribers. The following rules apply to adjustment control profiles:

- If no adjustment control profile is configured, the default adjustment control profile is used.
- You can configure a maximum of one adjustment control profile; a commit error occurs if you configure more than one adjustment control profile.
- If an application is not configured with an adjustment control profile, Junos OS uses its default values for priority and algorithm. For example, if ANCP is not configured in the adjustment control profile, the ANCP application is set to a priority of 1 and the algorithm is set to adjust-always.
- Adjustment control profiles apply to both static and dynamic interfaces.
- You can configure the algorithm to the following values:

  NOTE: All values can apply to shaping rates. Only adjust-never and adjust always can apply to overhead-accounting attributes.

  - adjust-never—Do not perform rate adjustments.
  - adjust-always—Adjust the shaping rate unconditionally.
  - adjust-less—Adjust the shaping rate if it is less than the configured value.
  - adjust-less-or equal—Adjust the shaping rate if it is less than or equal to the configured value.
  - adjust-greater—Adjust the shaping rate if it is greater than the configured value.
  - adjust-greater-or-equal—Adjust the shaping rate if it is greater than or equal to the configured value.
- When you modify an adjustment control profile, the changes take effect immediately and the modified profile is used for all further adjustments. However, existing adjustments are not reevaluated when you modify the adjustment control profile.

For example, if you have an ANCP adjustment that overrides a PPPoE adjustment on interface ge-1/1/0.100, and then you use the adjustment control profile to change the priority so that the ANCP priority is now lower than the PPPoE priority, Junos OS does not go back and reevaluate the adjustment on ge-1/1/0.100.

CoS Shaping Rate Fallback Behavior

When a CoS service profile is deactivated or removed, the CoS shaping rate falls back to the next highest available adjustment source as follows:

1. Fall back to the ANCP shaping rate if it is present and it has a higher priority than RADIUS CoA, the DHCP tags, or the PPPoE IA tags.
2. Fall back to the RADIUS CoA shaping rate if it is present and it has a higher priority than the DHCP tags or the PPPoE IA tags.
3. Fall back to the DHCP tags or the PPPoE IA tags shaping rate, if present.
4. Fall back to the shaping rate configured in the associated traffic control profile.

When a shaping rate is adjusted by ANCP, if that adjustment is removed, the rate reverts to the PPPoE IA tag rate if it is present. If the tag rate is not present then the shaping rate reverts to the configured rate in the traffic control profile.

When an ANCP adjustment for overhead-accounting mode is removed, the value reverts to the PPPoE IA tag value if it is present. If the tag value is not present, then the mode reverts to the configured value in the traffic control profile.

When an ANCP adjustment for overhead-accounting bytes is removed, the value reverts to the configured value in the traffic control profile; PPPoE IA tags cannot provide this value.

RELATED DOCUMENTATION

Configuring CoS Adjustment Control Profiles | 192

Verifying the CoS Adjustment Control Profile Configuration | 194

Configuring CoS Adjustment Control Profiles

To configure adjustment control profiles:

NOTE: You can only configure one adjustment control profile.

1. Configure the adjustment control profile name.

```
[edit]
user@host#edit class-of-service adjustment-control-profiles profile-name
```

2. (Optional) Configure the adjustment controls for the Access Node Control Protocol (ANCP) application:

```
[edit class-of-service adjustment-control-profiles profile-name ]
user@host# set application ancp priority priority algorithm algorithm
```

3. (Optional) Configure the adjustment controls for the RADIUS CoA application:

```
[edit class-of-service adjustment-control-profiles profile-name ]
user@host# set application radius-coa priority priority algorithm algorithm
```

4. (Optional) Configure the adjustment controls for the PPPoE tags:

```
[edit class-of-service adjustment-control-profiles profile-name ]
user@host# set application pppoe-tags priority priority algorithm algorithm
```

5. (Optional) Configure the adjustment controls for the DHCP application.

```
[edit class-of-service adjustment-control-profiles profile-name ]
user@host# set application dhcp-tags priority priority algorithm algorithm
```

6. (Optional) Verify your configuration.

```
user@host> show class-of-service adjustment-control-profiles
name: ANCP, priority: 1, algorithm: less;
name: RADIUS CoA, priority: 1, algorithm: always;
name: PPPoE IA tags, priority: 2, algorithm: less;
name: DHCP tags, priority: 2, algorithm: less
```

RELATED DOCUMENTATION

CoS Adjustment Control Profiles Overview | 189

Verifying the CoS Adjustment Control Profile Configuration | 194

Verifying the CoS Adjustment Control Profile Configuration

IN THIS SECTION

- Purpose | 194
- Action | 194

Purpose

View the class-of-service (CoS) adjustment control profile.

Action

- To display the CoS adjustment control profile:

```
user@host>  show class-of-service adjustment-control-profile profile-name
```

```
user@host> show class-of-service adjustment-control-profile acp1
  name: ANCP, priority: 1, algorithm: less
  name: RADIUS CoA, priority: 1, algorithm: always
  name: PPPoE IA tags, priority: 2, algorithm: less
  name: DHCP tags, priority: 2, algorithm: less
```

RELATED DOCUMENTATION

CoS Adjustment Control Profiles Overview | 189

Configuring CoS Adjustment Control Profiles | 192

adjustment-control-profiles

application (Adjustment Control Profiles)

CHAPTER 13

Applying CoS to Groups of Subscriber Interfaces

IN THIS CHAPTER

- CoS for Interface Sets of Subscribers Overview | 196
- Configuring an Interface Set of Subscribers in a Dynamic Profile | 199
- Example: Configuring a Dynamic Interface Set of VLAN Subscribers | 200
- Example: Configure a Dynamic Service VLAN Interface Set of Subscribers in a Dynamic Profile | 221

CoS for Interface Sets of Subscribers Overview

IN THIS SECTION

- Guidelines for Configuring Dynamic Interface Sets in a Subscriber Access Network | 196

Interface sets enable service providers to group logical interfaces or other interface sets so they can apply CoS parameters to all of the traffic in the group.

Interface sets are beneficial for various scenarios in a subscriber access network. For example, you can use an interface set to configure a local loop with a small number of subscribers. Interface sets are also useful for grouping a large number of subscribers into a particular service class or for defining traffic engineering aggregates for DSLAMs.

Guidelines for Configuring Dynamic Interface Sets in a Subscriber Access Network

When configuring interface sets for subscriber access, keep the following guidelines in mind:

- You can configure interface sets of VLAN demux, PPPoE, or demux interfaces over aggregated Ethernet interfaces.

- An interface can only belong to one interface set. If you try to add the same interface to different interface sets, the commit operation fails.
- You configure the interface set and the traffic scheduling and shaping parameters in a dynamic profile. However, you must apply the traffic-control profile to the interface set in the static `[edit class-of-service]` hierarchy.

  NOTE: This rule applies to all interface sets except ACI sets.

- The `$junos-interface-set-name` predefined variable is available only for RADIUS Accept messages; change of authorization (CoA) requests are not supported.
- The `$junos-aggregation-interface-set-name` is the L2 interface-set representing a logical intermediate node (DPU-C or PON tree) in the access network.
- The `$junos-phy-ifd-underlying-intf-set-name` represents a default, topology-based interface-set (based on the physical interface name with a post-pend of "-underlying") to conserve L2 CoS nodes.
- The `$junos-svlan-interface-set-name` predefined variable locally generates an interface set name for use by dual-tagged VLAN interfaces based on the outer tag of the dual-tagged VLAN. The format of the generated variable is *physical_interface_name - outer_VLAN_tag*. For example, an aggregated Ethernet interface "ae0," with a dual-tagged VLAN interface that has an outer tag of "111," results in a `$junos-svlan-interface-set-name` dynamic variable of "ae0-111". Similarly, a non-aggregated Ethernet interface of ge-1/1/0, with the same dual-tagged VLAN interface that has an outer tag of "111," results in a `$junos-svlan-interface-set-name` dynamic variable of "ge-1/1/0-111".
- The `$junos-phy-ifd-interface-set-name` predefined variable locally generates an interface set name associated with the underlying physical interface in a dynamic profile. This predefined variable enables you to group all the subscribers on a specific physical interface so that you can apply services to the entire group of subscribers.

  Another use case for this predefined variable is to conserve CoS resources in a mixed business and residential topology by collecting the residential subscribers into an interface set associated with the physical interface, so that a level 2 node is used for the interface set rather than for each residential interface. Otherwise, because the business and residential subscribers share the same interface and business subscribers require three levels of CoS, then three levels are configured for each residential subscriber. That results in an unnecessary level 2 node being consumed for each residential connection, wasting CoS resources.
- The `$junos-tagged-vlan-interface-set-name` predefined variable locally generates an interface set name used for grouping logical interfaces stacked over logical stacked VLAN demux interfaces for either a 1:1 (dual-tagged; individual client) VLAN or N:1 (single tagged; service) VLAN. The format of the generated variable differs with VLAN type as follows:

  - Dual-tagged (client) VLAN—*physical_interface_name - outer_VLAN_tag - inner_VLAN_tag*. For example, an aggregated Ethernet interface “ae0,” with a dual-tagged VLAN interface that has an outer tag of “111” and an inner tag of “200,” results in a `$junos-tagged-vlan-interface-set-name` dynamic variable of “ae0-200-111”. Similarly, a non-aggregated Ethernet interface of ge-1/1/0, with the same dual-tagged VLAN interface that has an outer tag of “111” and an inner tag of “200,” results in a `$junos-tagged-vlan-interface-set-name` dynamic variable of “ge-1/1/0-200-111”.

  - Single tagged (service) VLAN—*physical_interface_name - VLAN_tag*. For example, an aggregated Ethernet interface “ae0,” with an N:1 VLAN using the single tag of “200,” results in a `$junos-tagged-vlan-interface-set-name` dynamic variable of “ae0-200”. Similarly, a non-aggregated Ethernet interface of ge-1/1/0, with the same N:1 VLAN using the single tag of “200,” results in a `$junos-tagged-vlan-interface-set-name` dynamic variable of “ge-1/1/0-200”.

- All dynamic demux, dual-tagged VLAN logical interfaces with the same outer VLAN tag and physical interface are assigned to the same interface set and all CoS values provisioned with the dynamic profile are applied to the interfaces that are part of the set.

- The interface set name must be explicitly referenced in the CoS configuration as part of the static configuration outside of the dynamic profile. The CoS configuration is static and the interface set name must be statically referenced.

  NOTE: This rule applies to all interface sets except ACI sets.

- RADIUS can return an *access-accept* message under certain conditions. A configured RADIUS VSA for the interface set name takes precedence over the locally generated variable on the router. This means that if the interface-set-name VSA is configured on RADIUS, the router continues to use this variable instead of the locally generated value from the dynamic variable.

- Sets of aggregated Ethernet interfaces are supported on MPC/MIC interfaces on MX Series routers only.

- The supported interface stacks for aggregated Ethernet in an interface set include VLAN demux interfaces, IP demux interfaces, and PPPoE logical interfaces over VLAN demux interfaces.

- The link membership list and scheduler mode of the interface set are inherited from the underlying aggregated Ethernet interface over which the interface set is configured.

- When an aggregated Ethernet interface operates in link protection mode, or if the scheduler mode is configured to replicate member links, the scheduling parameters of the interface set are copied to each of the member links.

- If the scheduler mode of the aggregated Ethernet interface is set to scale member links, the scheduling parameters are scaled based on the number of active member links and applied to each of the aggregated interface member links.

RELATED DOCUMENTATION

Configuring an Interface Set of Subscribers in a Dynamic Profile | 199

Configuring an Interface Set of Subscribers in a Dynamic Profile | 199

Configuring an Interface Set of Subscribers in a Dynamic Profile

Interface sets enable you to provide hierarchical scheduling to a group of subscriber interfaces.

Before you begin, configure the subscriber interfaces that you intend to include in the interface set.

To configure an interface set of subscriber interfaces:

1. Configure the interface set in the dynamic profile.

```
[edit dynamic-profiles profile-name interfaces]
user@host#edit interface-set interface-set-name
```

   Replacing the *interface-set-name* variable with the `$junos-interface-set-name`, `$junos-svlan-interface-set-name`, or `$junos-tagged-vlan-interface-set-name` predefined variable. The interface set is created dynamically when the subscriber logs in.

2. Include the interfaces within the dynamic interface-set.

```
[edit dynamic-profiles profile-name interfaces interface-set $junos-interface-set-name]
user@host# set interface interface-name unit logical-unit-number
```

3. Apply traffic shaping and queuing parameters to the interface set.

   TIP: You must configure the interface set in the static `[edit class-of-service]` hierarchy, not in the `[edit dynamic-profiles]` hierarchy.

```
[edit class-of-service interfaces]
user@host# edit interface-set interface-set-name
[edit class-of-service interfaces interface-set interface-set-name]
user@host# set output-traffic-control-profile profile-name
```

RELATED DOCUMENTATION

CoS for Interface Sets of Subscribers Overview | 196
Guidelines for Configuring Dynamic CoS for Subscriber Access | 44
CoS for Interface Sets of Subscribers Overview | 196
Example: Configuring a Dynamic Interface Set of VLAN Subscribers | 200
CoS for Aggregated Ethernet Subscriber Interfaces Overview | 49

Example: Configuring a Dynamic Interface Set of VLAN Subscribers

IN THIS SECTION

- Requirements | 200
- Overview | 200
- Configuring the Dynamic VLANs | 201
- Configuring Dynamic Traffic Scheduling and Shaping | 204
- Configuring the Interface Set in the Dynamic Profile | 209
- Configuring DHCP Access | 212
- Configuring RADIUS Authentication | 214
- Verification | 220

Requirements

This example uses the following software and hardware components:

- MX Series Router with MPCs

Overview

In this example, the network administrator groups dynamic VLAN interfaces in an interface set. The interface set is configured in a dynamic profile, and enables hierarchical scheduling for the VLAN interfaces for a multiplay service.

DHCP is used as the access method, and RADIUS is used as the authentication method for the interfaces associated with the interface set.

Configuring the Dynamic VLANs

IN THIS SECTION

- CLI Quick Configuration | 201
- Configuring the Dynamic Profile for the Autoconfigured VLANs | 201
- Configuring the VLAN Interfaces | 202
- Configuring the Loopback Interface | 203

CLI Quick Configuration

To quickly configure the dynamic VLANs, copy the following commands and paste them into the router terminal window:

```
[edit]
edit dynamic-profiles vlan-prof
edit interfaces $junos-interface-ifd-name unit $junos-interface-unit
set vlan-id $junos-vlan-id
set demux-source inet
set family inet unnumbered-address lo0.0 preferred-source-address 203.0.113.32
top
edit interfaces ge-1/0/0
set hierarchical-scheduler
set vlan-tagging
edit auto-configure vlan-ranges dynamic-profile vlan-prof
set ranges any
set accept inet
top
set interfaces lo0 unit 0 family inet address 203.0.113.32/32
```

Configuring the Dynamic Profile for the Autoconfigured VLANs

Step-by-Step Procedure

In this section, you create a dynamic profile for the VLAN IDs to be automatically assigned when subscribers log in.

To configure the dynamic profile for the VLANs:

1. Configure the dynamic profile.

```
[edit]
user@host#edit dynamic-profile vlan-prof
```

2. Configure the interfaces.

```
[edit dynamic-profiles vlan-prof]
user@host#edit interfaces $junos-interface-ifd-name unit $junos-interface-unit
```

3. Add the VLAN ID variable.

```
[edit dynamic-profiles vlan-prof interfaces $junos-interface-ifd-name unit $junos-interface-
unit]
user@host#set vlan-id $junos-vlan-id
```

4. Configure the demux source as IPv4.

```
[edit dynamic-profiles vlan-prof interfaces $junos-interface-ifd-name unit $junos-interface-
unit]
user@host#set demux-source inet
```

5. Configure the family.

```
[edit dynamic-profiles vlan-prof interfaces $junos-interface-ifd-name unit $junos-interface-
unit]
user@host#set family inet unnumbered-address lo0.0 preferred-source-address 203.0.113.32
```

Configuring the VLAN Interfaces

Step-by-Step Procedure

To configure the VLAN interfaces:

1. Create the VLAN interface.

```
[edit]
user@host# edit interfaces ge-1/0/0
```

2. Enable hierarchical scheduling.

```
[edit interfaces ge-1/0/0]
user@host# set hierarchical-scheduler
```

3. Configure VLAN tagging.

```
[edit interfaces ge-1/0/0]
user@host# set vlan-tagging
```

4. Configure auto-configuration for the dynamic profile.

```
[edit interfaces ge-1/0/0]
user@host# edit auto-configure vlan-ranges dynamic-profile vlan-prof
```

5. Configure any VLAN ID range.

```
[edit interfaces ge-1/0/0 auto-configure vlan-ranges dynamic-profile vlan-prof]
user@host# set ranges any
```

6. Specify IPv4 traffic for the VLAN.

```
[edit interfaces ge-1/0/0 auto-configure vlan-ranges dynamic-profile vlan-prof]
user@host# set accept inet
```

Configuring the Loopback Interface

Step-by-Step Procedure

To configure the loopback interface:

1. Create the loopback interface.

```
[edit]
user@host# edit interfaces lo0
```

2. Configure the unit and the family.

```
[edit intefaces lo0]
user@host# set unit 0 family inet address 203.0.113.32/32
```

Configuring Dynamic Traffic Scheduling and Shaping

IN THIS SECTION

- CLI Quick Configuration | 204
- Configuring the Schedulers in the Dynamic Profile | 206
- Configuring the Scheduler Map in the Dynamic Profile | 208
- Configuring the Traffic-Control Profile in the Dynamic Profile | 208

CLI Quick Configuration

To quickly configure the traffic scheduling and shaping parameters, copy the following commands and paste them into the router terminal window:

```
[edit]
edit dynamic-profiles multiplay class-of-service schedulers be_sch
set transmit-rate percent 12
set buffer-size percent 12
set priority low
up
edit ef_sch
set transmit-rate percent 12
set buffer-size percent 12
set priority low
up
edit af_sch
```

```
set transmit-rate percent 12
set buffer-size percent 12
set priority low
up
edit nc_sch
set transmit-rate percent 12
set buffer-size percent 12
set priority low
up
edit voice_sch
set transmit-rate percent 12
set buffer-size percent 12
set priority low
up
edit video_sch
set transmit-rate percent 12
set buffer-size percent 12
set priority low
up
edit game_sch
set transmit-rate percent 12
set buffer-size percent 12
set priority low
up
edit data_sch
set transmit-rate percent 12
set buffer-size percent 12
set priority low
up 2
edit scheduler-maps all_smap
set forwarding-class be scheduler be_sch
set forwarding-class ef scheduler ef_sch
set forwarding-class af scheduler af_sch
set forwarding-class nc scheduler nc_sch
set forwarding-class voice scheduler voice_sch
set forwarding-class video scheduler video_sch
set forwarding-class game scheduler game_sch
set forwarding-class data scheduler data_sch
up 2
edit traffic-control-profiles multiplay
set scheduler-map all_smap
```

```
set shaping-rate 100m
set guaranteed-rate 20m
```

Configuring the Schedulers in the Dynamic Profile

Step-by-Step Procedure

In this section, you create a dynamic profile for the multiplay service and configure scheduling and shaping.

To configure the schedulers:

1. Create the `multiplay` dynamic profile.

```
[edit]
user@host# edit dynamic-profiles multiplay class-of-service schedulers
```

2. Configure the best effort scheduler.

```
[edit dynamic-profiles multiplay class-of-service schedulers]
user@host# edit be_sch
user@host# set transmit-rate percent 12
user@host# set buffer-size percent 12
user@host# set priority low
```

3. Configure the expedited forwarding scheduler.

```
[edit dynamic-profiles multiplay class-of-service schedulers]
user@host# edit ef_sch
user@host# set transmit-rate percent 12
user@host# set buffer-size percent 12
user@host# set priority low
```

4. Configure the assured forwarding scheduler.

```
[edit dynamic-profiles multiplay class-of-service schedulers]
user@host# edit af_sch
user@host# set transmit-rate percent 12
```

```
user@host# set buffer-size percent 12
user@host# set priority low
```

5. Configure the network control scheduler.

```
[edit dynamic-profiles multiplay class-of-service schedulers]
user@host# edit nc_sch
user@host# set transmit-rate percent 12
user@host# set buffer-size percent 12
user@host# set priority low
```

6. Configure the voice scheduler.

```
[edit dynamic-profiles multiplay class-of-service schedulers]
user@host# edit voice_sch
user@host# set transmit-rate percent 12
user@host# set buffer-size percent 12
user@host# set priority low
```

7. Configure the video scheduler.

```
[edit dynamic-profiles multiplay class-of-service schedulers]
user@host# edit video_sch
user@host# set transmit-rate percent 12
user@host# set buffer-size percent 12
user@host# set priority low
```

8. Configure the gaming scheduler.

```
[edit dynamic-profiles multiplay class-of-service schedulers]
user@host# edit game_sch
user@host# set transmit-rate percent 12
user@host# set buffer-size percent 12
user@host# set priority low
```

9. Configure the data scheduler.

```
[edit dynamic-profiles multiplay class-of-service schedulers]
user@host# edit data_sch
user@host# set transmit-rate percent 12
user@host# set buffer-size percent 12
user@host# set priority low
```

Configuring the Scheduler Map in the Dynamic Profile

Step-by-Step Procedure

To configure the scheduler map:

1. Configure the scheduler map for all of the services.

```
[edit dynamic-profiles multiplay class-of-service]
user@host# edit scheduler-maps all_smap
```

2. Configure the forwarding classes for each service in the scheduler map.

```
[edit dynamic-profiles multiplay class-of-service scheduler-maps all_smap]
user@host# set forwarding-class be scheduler be_sch
user@host# set forwarding-class ef scheduler ef_sch
user@host# set forwarding-class af scheduler af_sch
user@host# set forwarding-class nc scheduler nc_sch
user@host# set forwarding-class voice scheduler voice_sch
user@host# set forwarding-class video scheduler video_sch
user@host# set forwarding-class game scheduler game_sch
user@host# set forwarding-class data scheduler data_sch
```

Configuring the Traffic-Control Profile in the Dynamic Profile

Step-by-Step Procedure

To configure the traffic-control profile the interface set:

1. Configure the traffic-control profile.

```
[edit dynamic-profiles multiplay class-of-service]
user@host# edit traffic control-profiles multiplay
```

2. Configure the scheduler map.

```
[edit dynamic-profiles multiplay class-of-service traffic control-profiles multiplay]
user@host# set scheduler-map all_smap
```

3. Configure the shaping rate.

```
[edit dynamic-profiles multiplay class-of-service traffic control-profiles multiplay]
user@host# set shaping-rate 100m
```

4. Configure the guaranteed rate.

```
[edit dynamic-profiles multiplay class-of-service traffic control-profiles multiplay]
user@host# set guaranteed-rate 20m
```

Configuring the Interface Set in the Dynamic Profile

IN THIS SECTION

- CLI Quick Configuration | 210
- Configuring the Interfaces for the Interface Set | 210
- Configuring the Interface Set | 211
- Applying the Traffic-Control Profile to the Interface Set | 211

CLI Quick Configuration

To quickly configure the interface set, copy the following commands and paste them into the router terminal window:

```
[edit]
edit dynamic-profiles multiplay
edit interfaces interface-set $junos-interface-set-name
set interface $junos-interface-ifd-name unit $junos-underlying-interface-unit
top
edit class-of-service interfaces interface-set
set output-traffic-control-profile multiplay
```

Configuring the Interfaces for the Interface Set

Step-by-Step Procedure

To configure the interface variable for the interface set:

1. Configure the dynamic profile for the interface set.

```
[edit]
user@host#edit dynamic-profiles multiplay
```

2. Configure the interface using the Junos OS predefined variable.

```
[edit dynamic-profiles multiplay]
user@host#edit interfaces $junos-interface-ifd-name unit $junos-underlying-interface-unit
```

3. Configure the family.

```
[edit dynamic-profiles multiplay interfaces $junos-interface-set-name unit $junos-underlying-
interface-unit]
user@host#set family inet unnumbered-address lo0.0 preferred-source-address 203.0.113.32
```

Configuring the Interface Set

Step-by-Step Procedure

To configure the interface set:

1. Configure the interface set using the Junos OS predefined variable.

```
[edit dynamic-profiles multiplay]
user@host#edit interfaces interface-set $junos-interface-set-name
```

2. Add the dynamic VLAN interfaces to the interface set.

```
[edit dynamic-profiles multiplay interfaces $junos-interface-set-name]
user@host#set interface $junos-interface-ifd-name unit $junos-underlying-interface-unit
```

Applying the Traffic-Control Profile to the Interface Set

Step-by-Step Procedure

You apply the traffic-control profile outside of the dynamic profile in the `[edit class-of-service]` hierarchy.

To apply the traffic-control profile:

1. Specify the interface set to which you want to apply the traffic-control profile.

```
[edit class-of-service]
user@host#edit interfaces interface-set dynamic-set
```

2. Attach the output traffic-control profile defined in the dynamic profile to the interface set.

```
[edit class-of-service interfaces]
user@host#set output-traffic-control-profile multiplay
```

Configuring DHCP Access

IN THIS SECTION

- CLI Quick Configuration | 212
- Configuring the DHCP Local Server | 212
- Configuring Address Assignment Pools | 213

CLI Quick Configuration

To quickly configure DHCP access, copy the following commands and paste them into the router terminal window:

```
[edit]
edit system services dhcp-local-server authentication
set password $ABC123
set username-include user-prefix multiplay
up 1
set dynamic-profile dhcp-vlan-prof aggregate-clients replace
set group vlans interface ge-1/0/0
top
edit access address-assignment pool v4 family inet
set network 203.0.113.0/16
set range limited low 203.0.113.10
set range limited high 203.0.113.250
set dhcp-attributes maximum-lease-time 84600
```

Configuring the DHCP Local Server

Step-by-Step Procedure

To configure DHCP access:

1. Configure the DHCP local server.

```
[edit system]
user@host# edit services dhcp-local-server authentication
```

2. Set the password.

```
[edit system services dhcp-local-server authentication]
user@host# set password $ABC123
```

3. Specify that you want to include optional information in the username.

```
[edit system services dhcp-local-server authentication]
user@host# set username-include user-prefix multiplay
```

4. Attach the dynamic profile with the interface set.

```
[edit system services dhcp-local-server]
user@host# set dynamic-profile dhcp-vlan-prof aggregate-clients replace
```

5. Configure a group for the VLAN interface.

```
[edit system services dhcp-local-server]
user@host# set group vlans interface ge-1/0/0
```

Configuring Address Assignment Pools

Step-by-Step Procedure

To configure address assignment pools:

1. Configure the pool of IPv4 addresses.

```
[edit access]
user@host#edit address-assignment pool v4 family inet
```

2. Configure the family of interfaces in the pool.

```
[edit access address-assignment pool v4]
user@host#set network 203.0.113.0/16
```

3. Configure the upper and lower bounds of the address range.

```
[edit access address-assignment pool v4]
user@host#set range limited low 203.0.113.10
user@host#set range limited high 203.0.113.250
```

4. Configure the maximum length of time in seconds for which a subscriber can request and hold a lease.

```
[edit access address-assignment pool v4]
user@host#set dhcp-attributes maximum-lease-time 84600
```

Configuring RADIUS Authentication

IN THIS SECTION

- CLI Quick Configuration | 214
- Configuring RADIUS Access | 215
- Results | 216

CLI Quick Configuration

To quickly configure RADIUS authentication, copy the following commands and paste them into the router terminal window:

```
[edit]
edit access radius-server 192.51.100.108
set secret $ABC123ABC123ABC123
set timeout 5
set retry 5
up 2
edit profile acc-prof
set authentication-order radius
set radius authentication-server 192.51.100.108
```

Configuring RADIUS Access

Step-by-Step Procedure

To configure RADIUS access:

1. Configure the RADIUS server.

```
[edit access]
user@host#edit radius-server 192.51.100.108
```

2. Configure the required secret (password) that the local router or switch passes to the RADIUS client.

```
[edit access radius-server 192.51.100.108]
user@host# set secret $ABC123ABC123ABC123
```

3. Configure the length of time that the local router or switch waits to receive a response from a RADIUS server.

```
[edit access radius-server 192.51.100.108]
user@host# set timeout 5
```

4. Configure the number of times that the router or switch attempts to contact a RADIUS accounting server.

```
[edit access radius-server 192.51.100.108]
user@host# set retry 5
```

5. Configure the access profile.

```
[edit access]
user@host#edit profile acc-prof
```

6. Configure the authentication order.

```
[edit access profile acc-prof ]
user@host# set authentication-order radius
```

7. Configure the authentication server.

```
[edit access profile acc-prof]
user@host#set radius authentication-server 192.51.100.108
```

Results

```
dynamic-profiles {
    vlan-prof {
        interfaces {
            “$junos-interface-ifd-name” {
                unit "$junos-interface-unit" {
                    vlan-id "$junos-vlan-id";
                    demux-source inet;
                    family inet {
                        unnumbered-address lo0.0 preferred-source-address 203.0.113.32;
                    }
                }
            }
        }
    }
    multiplay {
        class-of-service {
            traffic-control-profiles {
                multiplay {
                    scheduler-map all_smap;
                    shaping-rate 100m;
                    guaranteed-rate 20m;
                }
            }
            interfaces {
                    interface-set “$junos-interface-set-name” {
                        interface “$junos-interface-ifd-name” {
                            unit “$junos-underlying-interface-unit”;
                        }
                    }
                “$junos-interface-ifd-name” {
                    unit "$junos-interface-unit" {
                        output-traffic-control-profile multiplay;
                    }
                }
```

```
        }
        scheduler-maps {
            all_smap {
                forwarding-class be scheduler be_sch;
                forwarding-class ef scheduler ef_sch;
                forwarding-class af scheduler af_sch;
                forwarding-class nc scheduler nc_sch;
                forwarding-class voice scheduler voice_sch;
                forwarding-class video scheduler video_sch;
                forwarding-class game scheduler game_sch;
                forwarding-class data scheduler data_sch;
            }
        }
        schedulers {
            be_sch {
                transmit-rate percent 12;
                buffer-size percent 12;
                priority low;
            }
            ef_sch {
                transmit-rate percent 12;
                buffer-size percent 12;
                priority low;
            }
            af_sch {
                transmit-rate percent 12;
                buffer-size percent 12;
                priority low;
            }
            nc_sch {
                transmit-rate percent 12;
                buffer-size percent 12;
                priority low;
            }
            voice_sch {
                transmit-rate percent 12;
                buffer-size percent 12;
                priority low;
            }
            video_sch {
                transmit-rate percent 12;
                buffer-size percent 12;
                priority low;
```

```
                }
                game_sch {
                    transmit-rate percent 12;
                    buffer-size percent 12;
                    priority low;
                }
                data_sch {
                    transmit-rate percent 12;
                    buffer-size percent 12;
                    priority low;
                }
            }
            }
        }
        access {
            radius-server {
                192.51.100.108 {
                    secret "$ABC123ABC123ABC123"; ## SECRET-DATA
                    timeout 5;
                    retry 5;
                }
            }
            profile acc-prof {
                authentication-order radius;
                radius {
                    authentication-server 192.51.100.108;
                }
            }
            address-assignment {
                pool v4 {
                    family inet {
                        network 203.0.113.0/16;
                        range limited {
                            low 203.0.113.10;
                            high 203.0.113.250;
                        }
                        dhcp-attributes {
                            maximum-lease-time 84600;
                        }
                    }
                }
            }
        }
```

```
        class-of-service {
            interfaces {
                interface-set dynamic-set {
                    output-traffic-control-profile multiplay;
                }
            }
        }
        interfaces {
            interface-set “$junos-interface-set-name” {
                interface "$junos-interface-ifd-name" {
                    unit "$junos-underlying-interface-unit";
                }
            }
            "$junos-interface-ifd-name" {
                unit "$junos-underlying-interface-unit" {
                    family inet {
                        unnumbered-address lo0.0 preferred-source-address 203.0.113.32;
                    }
                }
            }
        }
    }
}
    interfaces {
        ge-1/0/0 {
            hierarchical-scheduler;
            vlan-tagging;
            auto-configure {
                vlan-ranges {
                    dynamic-profile vlan-prof {
                        accept inet;
                        ranges {
                            any;
                        }
                    }
                }
            }
        }
        lo0 {
            unit 0 {
                family inet {
                    address 203.0.113.32/32;
                }
```

```
            }
        }
    }
    system {
        services {
            dhcp-local-server {
                authentication {
                    password $ABC123;
                    username-include {
                        user-prefix multiplay;
                    }
                }
                dynamic-profile multiplay aggregate-clients replace;
                group vlans {
                    interface ge-1/0/0.0;
                }
            }
        }
    }
```

Verification

IN THIS SECTION

- Verifying the Interfaces that are Included in the Interface Set | **220**
- Verifying the Traffic Scheduling and Shaping Parameters for the Interface Set | **221**

To confirm that the configuration is correct, perform these tasks:

Verifying the Interfaces that are Included in the Interface Set

Purpose

Verify the interfaces included in the interface set.

Action

```
user@host> show interfaces interface-set dynamic-set terse
```

Verifying the Traffic Scheduling and Shaping Parameters for the Interface Set

Purpose

Verify that the traffic scheduling and shaping parameters are applied properly to an interface included in the interface set.

Action

```
user@host> show class-of-service interface
```

RELATED DOCUMENTATION

- Understanding Hierarchical CoS for Subscriber Interfaces
- Configuring an Interface Set of Subscribers in a Dynamic Profile | 199

Example: Configure a Dynamic Service VLAN Interface Set of Subscribers in a Dynamic Profile

IN THIS SECTION

- Requirements | 222
- Overview | 222
- Configuration | 223
- Verification | 226

Interface sets enable you to provide hierarchical scheduling to a group of subscriber interfaces. In this example, by using the `$junos-svlan-interface-set-name` internal dynamic variable when specifying the interface set name, you can locally generate an interface set name for use by SVLAN interfaces based on the outer tag of the dual-tagged VLAN. The format of the generated variable is *physical_interface_name - outer_VLAN_tag*.

Requirements

Before you begin, configure the subscriber interfaces that you intend to include in the interface set. You can find general configuration instructions for the supported dynamic interface configuration in *DHCP Subscriber Interface Overview* and in the following:

- For dynamic VLAN interfaces, see *Configuring a Static or Dynamic VLAN Subscriber Interface over Aggregated Ethernet*.
- For dynamic IP demux interfaces, see *Configuring a Static or Dynamic IP Demux Subscriber Interface over Aggregated Ethernet*.
- For dynamic VLAN demux interfaces, see *Configuring Dynamic Subscriber Interfaces Using VLAN Demux Interfaces in Dynamic Profiles*.

Overview

Interface sets enable you to provide hierarchical scheduling to a group of subscriber interfaces. By using the `$junos-svlan-interface-set-name` internal dynamic variable when specifying the interface set name, you can locally generate an interface set name for use by SVLAN interfaces based on the outer tag of the dual-tagged VLAN. The format of the generated variable is *physical_interface_name - outer_VLAN_tag*.

This example includes the following statements:

- `interface-set`—Configures the name of the scheduler for dynamic CoS. In this example, you use the `$junos-svlan-interface-set-name` variable to obtain the locally generated interface set name for use by SVLAN interfaces based on the outer tag of the dual-tagged VLAN.
- `output-traffic-control-profile`—Applies an output traffic scheduling and shaping profile to the interface set.
- `output-traffic-control-profile-remaining`—Applies an output traffic scheduling and shaping profile for remaining traffic to the interface set.

Configuration

IN THIS SECTION

- CLI Quick Configuration | 223
- Procedure | 223
- Results | 225

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.

```
[edit]
set dynamic-profiles profile-dhcp-ipdemux interfaces interface-set $junos-svlan-interface-set-
name interface $junos-interface-ifd-name unit $junos-underlying-interface-unit
set dynamic-profiles profile-dhcp-ipdemux interfaces $junos-interface-ifd-name unit $junos-
underlying-interface-unit
set class-of-service traffic-control-profiles tcp1 scheduler-map schedMap
set class-of-service traffic-control-profiles tcp1 shaping-rate 50m
set class-of-service traffic-control-profiles tcp1 guaranteed-rate 200k
set class-of-service traffic-control-profiles tcp3 scheduler-map ss1q0q1
set class-of-service traffic-control-profiles tcp3 shaping-rate 20m
set class-of-service traffic-control-profiles tcp3 guaranteed-rate 5m
set class-of-service interfaces interface-set ae0-111 output-traffic-control-profile tcp1
set class-of-service interfaces interface-set ae0-111 output-traffic-control-profile-remaining
tcp3
```

Procedure

Step-by-Step Procedure

To configure an SVLAN interface set of subscriber interfaces:

1. Access the dynamic profile you want to modify for interface sets.

```
[edit]
user@host# edit dynamic-profiles profile-dhcp-ipdemux
```

2. Access the dynamic profile interface configuration.

```
[edit dynamic-profiles profile-dhcp-ipdemux]
user@host# edit interfaces
```

3. Configure the SVLAN interface set in the dynamic profile. The interface set is created dynamically when the subscriber logs in.

```
[edit dynamic-profiles profile-dhcp-ipdemux interfaces]
user@host# edit interface-set $junos-svlan-interface-set-name
```

4. Include dynamic IP demux interface creation within the dynamic interface set.

```
[edit dynamic-profiles profile-dhcp-ipdemux interfaces interface-set $junos-svlan-interface-
set-name]
user@host# set interface $junos-interface-ifd-name unit $junos-underlying-interface-unit
```

5. Access the SVLAN interface set name that you expect `$junos-svlan-interface-set-name` to generate. For example, to specify the expected interface set name for aggregated Ethernet interface ae0 and outer VLAN tag 111, include **ae0-111** for the *interface-set-name* variable.

```
[edit class-of-service interfaces]
user@host# edit interface-set ae0-111
```

6. Apply traffic shaping and queuing parameters to the SVLAN interface set.

TIP: You must configure the interface set in the static `[edit class-of-service]` hierarchy, not in the `[edit dynamic-profiles]` hierarchy.

```
[edit class-of-service interfaces interface-set ae0-111]
user@host# set output-traffic-control-profile tcp1
```

7. Apply traffic shaping and queuing parameters to any remaining traffic on the SVLAN interface set.

```
[edit class-of-service interfaces interface-set ae0-111]
user@host# set output-traffic-control-profile-remaining tcp3
```

Results

From configuration mode, confirm your configuration by entering the `show dynamic-profiles` command and the `show class-of-service` command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show dynamic-profiles
dynamic-profiles {
    profile-dhcp-ipdemux {
        interfaces {
            interface-set "$junos-svlan-interface-set-name" {
                interface "$junos-interface-ifd-name" {
                    unit "$junos-underlying-interface-unit";
                }
            }
            "$junos-interface-ifd-name" {
                unit "$junos-underlying-interface-unit";
            }
        }
    }
}
```

```
user@host# show class-of-service
class-of-service {
    traffic-control-profiles {
        tcp1 {
```

```
            scheduler-map schedMap;
            shaping-rate 50m;
            guaranteed-rate 200k;
        }
        tcp3 {
            inactive: scheduler-map ss1q0q1;
            shaping-rate 20m;
            guaranteed-rate 5m;
        }
    }
    interfaces {
        interface-set ae0-111 {
            output-traffic-control-profile tcp1;
            output-traffic-control-profile-remaining tcp3;
        }
    }
}
```

Verification

IN THIS SECTION

- Verifying the Interfaces that are Included in the Interface Set | 226
- Displaying Information for Active Subscribers | 227

To confirm that the configuration is correct, perform these tasks:

Verifying the Interfaces that are Included in the Interface Set

Purpose

Verify the interfaces that are included in the interface set.

Action

```
user@host> show class-of-service interface-set
```

Displaying Information for Active Subscribers

Purpose

Display information for active subscribers.

Action

```
user@host> show subscribers detail
```

RELATED DOCUMENTATION

- *Dynamic Profiles Overview*
- *Configuring a Basic Dynamic Profile*
- Configuring Hierarchical Schedulers for CoS
- Configuring Remaining Common Queues on MIC and MPC Interfaces | **111**

CHAPTER 14

Applying CoS to Subscriber Interfaces

IN THIS CHAPTER

- Applying Traffic Shaping and Scheduling to a Subscriber Interface in a Dynamic Profile | 228
- Applying Minimal Shaping and Scheduling to Remaining Subscriber Traffic | 230
- Applying a Rewrite Rule Definition to a Subscriber Interface in a Dynamic Profile | 231
- Applying a Classifier to a Subscriber Interface in a Dynamic Profile | 232

Applying Traffic Shaping and Scheduling to a Subscriber Interface in a Dynamic Profile

After you configure the traffic shaping and scheduling CoS parameters in a dynamic profile, you apply them to an interface. The output traffic-control profile enables you to provide traffic scheduling to the interface.

To apply CoS attributes to an interface in a dynamic profile:

1. Specify that you want to apply CoS attributes to an interface in the dynamic profile.

```
[edit]
set dynamic-profiles profile-name class-of-service
```

2. Configure the interface name and logical interface using a variable.

```
[edit dynamic-profiles profile-name class-of-service]
set interfaces $junos-interface-ifd-name unit $junos-underlying-interface-unit
```

3. (Only for Junos OS Evolved) Include the `service-accounting` option to enable the packet forwarding engine to report the queue counter statistics.

```
[edit dynamic-profiles profile-name class-of-service]
set interfaces $junos-interface-ifd-name unit $junos-underlying-interface-unit service-
accounting
```

4. Apply the output traffic-control profile to the interface.

```
[edit dynamic-profiles profile-name class-of-service]
set interfaces $junos-interface-ifd-name unit $junos-underlying-interface-unit output-traffic-
control-profile profile-name
```

You can use one of the following methods to specify the output traffic-control profile you want to use:

- Reference the `$junos-cos-traffic-control-profile` predefined variable. At subscriber login, subscriber management takes one of the following actions, in the order listed:

  a. If RADIUS is being used and it returns a value for the traffic-control profile, subscriber management uses the RADIUS value.

  b. If RADIUS is not being used, subscriber management uses the default traffic-control profile (which is specified by the `predefined-variables-default` statement at the `[edit dynamic-profiles]` hierarchy).

  For example:

```
[edit dynamic-profiles profile-name class-of-service]
set interfaces $junos-interface-ifd-name unit $junos-underlying-interface-unit output-
traffic-control-profile $junos-cos-traffic-control-profile
```

- Explicitly reference the name of the traffic-control profile.

  For example:

```
[edit dynamic-profiles profile-name class-of-service]
set interfaces $junos-interface-ifd-name unit $junos-underlying-interface-unit output-
traffic-control-profile tcp-sales-2
```

RELATED DOCUMENTATION

- Guidelines for Configuring Dynamic CoS for Subscriber Access | 44
- Configuring Static Hierarchical Scheduling in a Dynamic Profile
- Example: Maintaining a Constant Traffic Flow by Configuring a Static VLAN Interface with a Dynamic Profile for Subscriber Access
- Example: Configuring Dynamic Hierarchical Scheduling for Subscribers
- Verifying the Scheduling and Shaping Configuration for Subscriber Access | 72
- CoS for Subscriber Access Overview | 43

Applying Minimal Shaping and Scheduling to Remaining Subscriber Traffic

It is beneficial to apply a remaining traffic-control profile to a logical interface to provide minimal CoS scheduling when you have not configured or over-provisioned Layer 3 schedulers. In the event that schedulers are not available, the remaining subscriber traffic receives the essential level of service.

To configure scheduling for remaining subscriber traffic:

1. Enable hierarchical scheduling for the interface.

```
[edit interfaces interface-name]
user@host# set hierarchical-scheduler
```

2. Apply the remaining traffic-control profile to the port on which you enabled hierarchical scheduling.

```
[edit class-of-service interfaces interface-name]
user@host# set output-traffic-control-profile-remaining profile-name
```

RELATED DOCUMENTATION

- Applying Traffic Shaping and Scheduling to a Subscriber Interface in a Dynamic Profile | 228

Applying a Rewrite Rule Definition to a Subscriber Interface in a Dynamic Profile

Rewrite rules define the marking for various CoS values, including DSCP, DSCP IPv6, IP precedence, and IEEE 802.1 CoS values. Rewrite rules have an associated forwarding class and code-point alias or bit set.

NOTE: By default, subscriber lawful intercept does not intercept DHCP control packets that are generated by the routing engine. To ensure that a DHCP control packet generated by the routing engine is intercepted, you need to configure the ieee-802.1 rewrite-rule for VLAN demux.

For dynamic CoS, you define the rewrite rules mapping for the CoS values statically, then reference the rewrite rule configuration in the dynamic profile for the subscriber interface.

To configure a rewrite rule in a dynamic profile:

1. Define the rewrite-rules mapping for the traffic that passes through all queues on the interface. The available rewrite-rules types for dynamic CoS are `dscp`, `dscpv6`, `ieee-802.1` and `inet-precedence`.

   See Configuring Rewrite Rules.
2. Apply the rewrite-rules definition to the subscriber interface in the dynamic profile.

```
[edit dynamic-profiles profile-name class-of-service interfaces interface-name unit logical-unit-number]
user@host# edit rewrite-rules
```

3. Configure the applicable rewrite rule markers in the dynamic profile.
   - For DSCP:

```
[edit dynamic-profiles profile-name class-of-service interfaces interface-name unit logical-unit-number rewrite-rules]
user@host# set dscp (rewrite-name | default)
```

   - For DSCPv6:

```
[edit dynamic-profiles profile-name class-of-service interfaces interface-name unit logical-unit-number rewrite-rules]
user@host# set dscp-ipv6 (rewrite-name | default)
```

- For IEEE 802.1:

```
[edit dynamic-profiles profile-name class-of-service interfaces interface-name unit
logical-unit-number rewrite-rules]
user@host# set ieee-802.1 (rewrite-name | default) vlan-tag (outer | outer-and-inner)
```

- For inet-precedence:

```
[edit dynamic-profiles profile-name class-of-service interfaces interface-name unit
logical-unit-number rewrite-rules]
user@host# set inet-precedence (rewrite-name | default)
```

RELATED DOCUMENTATION

Guidelines for Configuring Dynamic CoS for Subscriber Access | 44

Example: Configuring Dynamic Hierarchical Scheduling for Subscribers

Verifying the Scheduling and Shaping Configuration for Subscriber Access | 72

Applying a Classifier to a Subscriber Interface in a Dynamic Profile | 232

Applying IEEE 802.1p Rewrite Rules to Dual VLAN Tags

Rewriting Packet Headers to Ensure Forwarding Behavior

Applying a Classifier to a Subscriber Interface in a Dynamic Profile

You can apply the classification map to a subscriber interface in a dynamic profile.

For dynamic CoS, you define the classification map for the CoS values statically, then reference the classifier configuration in the dynamic profile for the subscriber interface.

To apply a classifier to an interface in a dynamic profile:

1. Define the classifier.

   The available classifier types for dynamic CoS are `dscp`, `dscp-ipv6`, `ieee-802.1`, and `inet-precedence`.

   See Configuring Behavior Aggregate Classifiers.

2. Apply the classifier definition to the subscriber interface in the dynamic profile.

```
[edit dynamic-profiles profile-name class-of-service interfaces interface-name unit logical-unit-number]
user@host# edit classifiers
```

3. Configure the applicable classifiers in the dynamic profile.

   - For DSCP:

```
[edit dynamic-profiles profile-name class-of-service interfaces interface-name unit logical-unit-number classifiers]
user@host# set dscp (classifier-name | default)
```

   - For DSCPv6:

```
[edit dynamic-profiles profile-name class-of-service interfaces interface-name unit logical-unit-number classifiers]
user@host# set dscp-ipv6 (classifier-name | default)
```

   - For IEEE 802.1:

```
[edit dynamic-profiles profile-name class-of-service interfaces interface-name unit logical-unit-number classifiers]
user@host# set ieee-802.1 (classifier-name | default) vlan-tag (inner | outer)
```

   - For inet-precedence:

```
[edit dynamic-profiles profile-name class-of-service interfaces interface-name unit logical-unit-number classifiers]
user@host# set inet-precedence (classifier-name | default)
```

RELATED DOCUMENTATION

Guidelines for Configuring Dynamic CoS for Subscriber Access | 44

Example: Configuring Dynamic Hierarchical Scheduling for Subscribers

Verifying the Scheduling and Shaping Configuration for Subscriber Access | 72

Applying a Rewrite Rule Definition to a Subscriber Interface in a Dynamic Profile | 231

Understanding How Behavior Aggregate Classifiers Prioritize Trusted Traffic

3

PART

Configuring Dynamic Filters and Policers

- Dynamic Firewall Filters Overview | **236**
- Configuring Static Firewall Filters That Are Dynamically Applied | **239**
- Streamlining Processing of Chains of Static Filters | **248**
- Dynamically Attaching Static or Fast Update Filters to an Interface | **255**
- Configuring Filters That Are Created Dynamically | **258**
- Using Ascend Data Filters to Implement Firewalls Based on RADIUS Attributes | **321**
- Configuring Fast Update Filters to Provide More Efficient Processing Over Classic Static Filters | **344**
- Defending Against DoS and DDoS Attacks Using Unicast RPF and Fail Filters | **369**
- Improving Scaling and Performance of Filters on Static Subscriber Interfaces | **381**
- Configuring Dynamic Service Sets | **386**
- Configuring Rate-Limiting Premium and Non-Premium Traffic on an Interface Using Hierarchical Policers | **390**
- Configure Rate-Limiting Traffic on an Interface Using Enhanced Hierarchical Policers | **412**
- Monitoring and Managing Firewalls for Subscriber Access | **426**

CHAPTER 15

Dynamic Firewall Filters Overview

IN THIS CHAPTER

- Understanding Dynamic Firewall Filters | 236
- Defining Dynamic Filter Processing Order | 237

Understanding Dynamic Firewall Filters

Firewall filters provide rules that define whether to accept or reject packets that are transiting an interface on a router. The subscriber management feature supports four categories of firewall filters:

- Classic filters are static filters that are applied to an interface dynamically. They are compiled at commit time and then, when a service is activated, an interface-specific filter is created and attached to a *logical interface*. This dynamic application is performed by associating input or output filters with a dynamic profile. When triggered, a dynamic profile applies the filter to an interface. Because classic filters are static, they cannot contain subscriber-specific terms (also called rules).

- Parameterized filters allow you to implement customized filters for each subscriber session. In parameterized filters, you use variables to define a filter. When services are activated for a subscriber, actual values such as policing rates, destination addresses, or ports are substituted for the variables and are used to create filters.

- Ascend-Data-Filters allow you to create dynamic filters based on values received from the RADIUS server in the Ascend-Data-Filter attribute (RADIUS attribute 242). The filter is configured on the RADIUS server and contains rules that specifically match conditions for traffic and define an action for the router to perform. When services are activated for a subscriber, a filter is created based on the values in the RADIUS attribute. You can also use Ascend-Data-Filters to create static filters by configuring the Ascend-Data-Filter attribute in a dynamic profile.

- Fast update filters are similar to classic filters. However, fast update filters support subscriber-specific, rather than interface-specific, filter values. Fast update filters also allow individual filter terms to be incrementally added or removed from filters without requiring that the entire filter be recompiled for each modification. Fast update filters are essential for networking environments in which multiple subscribers share the same logical interface.

You configure firewall filters to determine whether to accept or reject traffic before it enters or exits an interface to which the *firewall filter* is applied. An *input* (or *ingress*) firewall filter is applied to packets that are entering a network. An *output* (or *egress*) firewall filter is applied to packets that are exiting a network. You can configure firewall filters to subject packets to filtering or class-of-service (CoS) marking (grouping similar types of traffic together and treating each type of traffic as a class with its own level of service priority).

RELATED DOCUMENTATION

- Classic Filters Overview | 239
- Ascend-Data-Filter Policies for Subscriber Management Overview | 321
- Parameterized Filters Overview | 258
- Fast Update Filters Overview | 344
- Dynamically Attaching Statically Created Filters for Any Interface Type | 256
- Dynamically Attaching Statically Created Filters for a Specific Interface Family Type | 255
- Dynamically Attaching Filters Using RADIUS Variables | 307

Defining Dynamic Filter Processing Order

You can force filter processing to occur in a particular order by using the `precedence` statement. You specify a precedence for input and output filters within a dynamic profile at the `[edit dynamic-profiles` *`profile-name`* `interfaces (`*`interface-name`* `| demux0) unit` *`logical-unit-number`* `family` *`family`*`]` hierarchy level.

The precedence range is from 0 through 250. Setting a lower precedence value for a filter gives it a higher precedence within the dynamic profile. A precedence of zero (the default) gives the filter the highest precedence. If no precedence is specified, the filter receives a precedence of zero (highest precedence). Filters with matching precedence (zero or otherwise) are applied in random order.

Before you define a precedence for a filter in a dynamic profile.

1. Create the filters you want to attach to the dynamic profile.

   See Firewall Filters Overview for information about firewall filters and how to create them.

2. Create a basic dynamic profile.

   See *Configuring a Basic Dynamic Profile*.

3. Attach the filters to the dynamic profile.

See "Dynamically Attaching Statically Created Filters for Any Interface Type" on page 256, "Dynamically Attaching Statically Created Filters for a Specific Interface Family Type" on page 255, or "Dynamically Attaching Filters Using RADIUS Variables" on page 307.

To define a precedence for an input and output filter:

1. Specify the input filter precedence in the dynamic profile.

```
[edit dynamic-profiles profile-name interfaces interface-name unit logical-unit-number family family]
user@host# set filter input precedence 50
```

2. Specify the output filter precedence in the dynamic profile.

```
[edit dynamic-profiles profile-name interfaces interface-name unit logical-unit-number family family]
user@host# set filter output precedence 5
```

RELATED DOCUMENTATION

Classic Filters Overview | 239

Firewall Filters Overview

CHAPTER 16

Configuring Static Firewall Filters That Are Dynamically Applied

IN THIS CHAPTER

- Classic Filters Overview | 239
- Basic Classic Filter Syntax | 243
- Examples: Configuring Static Filters | 244

Classic Filters Overview

IN THIS SECTION

- Classic Filter Types | 239
- Classic Filter Components | 240
- Classic Filter Processing | 241
- Guidelines for Creating and Applying Classic Filters for Subscriber Interfaces | 242

The dynamic firewall feature supports classic filters, which are static filters that are applied to an interface dynamically. They are compiled at commit time and then, when a service is activated, an interface-specific clone of the filter is created and attached to a *logical interface*. This dynamic application is performed by associating input or output filters with a dynamic profile.

This overview covers:

Classic Filter Types

The following classic filter types are supported:

- Port (Layer 2) *firewall filter*—Port firewall filters apply to Layer 2 switch ports. You can apply port firewall filters only in the ingress direction on a physical port.
- VLAN firewall filter—VLAN firewall filters provide access control for packets that enter a VLAN, are bridged within a VLAN, and leave a VLAN. You can apply VLAN firewall filters in both ingress and egress directions on a VLAN. VLAN firewall filters are applied to all packets that are forwarded to or forwarded from the VLAN.
- Router (Layer 3) firewall filter—You can apply a router firewall filter in both ingress and egress directions on Layer 3 (routed) interfaces.

Classic Filter Components

When creating a classic filter, you first define the family address type (`inet` or `inet6`) and then you define one or more terms that specify the filtering criteria and the action to take when a match occurs.

Each term, or rule, consists of the following components:

- Match conditions—Specify values or fields that the packet must contain. You can define various match conditions, including:
  - IP source address field
  - IP destination address field
  - Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) source port field
  - IP protocol field
  - Internet Control Message Protocol (ICMP) packet type
  - TCP flags
  - Interfaces

  Match conditions (Junos OS Evolved)—Specify values or fields that the packet must contain. You can define various match conditions, including:
  - Destination-prefix-list
  - Source-prefix-list
  - Forwarding-class-except
  - Destination-address
  - Destination-port
  - Source-port

  - Protocol

- Actions—Specifies what to do when a match condition occurs. Possible actions are to accept or discard a packet. In addition, packets can be counted to collect statistical information. If no action is specified for a term, the default action is to accept the packet.

  Actions (Junos OS Evolved)—accept, count, discard, enhanced-hierarchical-policer, forwarding-class, sample, skip (used only for service filter for dropping the packets), service (used only for service filter).

- Subscriber management firewall filters – Specifies static filters that are applied to a subscriber interface dynamically (Starting in Junos OS Evolved Release 24.2R1, additional filters, supported for ACX7100-48L, ACX7332, and ACX7348 devices)

- Forwarding class

- Loss priority

- Packet length

- DSCP

- Traffic class

Classic Filter Processing

The order of the terms within a classic filter is important. Packets are tested against each term in the order in which the terms are listed in the firewall filter configuration. When a firewall filter contains multiple terms, the router takes a top-down approach and compares a packet against the first term in the firewall filter. If the packet matches the first term, the router executes the action defined by that term to either accept or reject the packet, and no other terms are evaluated. If the router does not find a match between the packet and first term, it then compares the packet to the next term in the firewall filter by using the same match process. If no match occurs between the packet and the second term, the router continues to compare the packet to each successive term defined in the firewall filter until a match is found. If a packet does not match any terms in a firewall filter, the default action is to discard the packet.

You can also specify a precedence (from 0 through 255) for input and output filters within a dynamic profile to force filter processing in a particular order. Setting a lower precedence value for a filter gives it a higher precedence within the dynamic profile. Filters with lower precedence values are applied to interfaces before filters with higher precedence values. A precedence of zero (the default) gives the filter the highest precedence. If no precedence is specified, the filter receives a precedence of zero (highest precedence). Filters with matching precedence (zero or otherwise) are applied in random order.

NOTE: Dynamic filters do not process outbound packets that are sourced from the routing engine. To filter outbound packets that are sourced from the routing engine, you can create static outbound filters for each interface.

Guidelines for Creating and Applying Classic Filters for Subscriber Interfaces

Dynamic configuration of firewall filters is supported. However, you can also continue to create static firewall filters for interfaces as you do normally, and then dynamically apply those filters to statically created interfaces using dynamic profiles. You can also use dynamic profiles to attach input and output filters through RADIUS.

When creating and applying filters, keep the following in mind:

- Dynamic application of only input and output filters is supported.
- The filters must be interface-specific.
- You can create family-specific `inet` and `inet6` filters.
- You can create interface-specific filters at the `unit` level that apply to any family type (`inet` or `inet6`) configured on the interface.
- You can add or remove both IPv4 and IPv6 filters with the same service activation or deactivation.
- You can remove one filter type without impacting the other type of filter. For example, you can remove IPv6 filters and leave the current IPv4 filters active.
- You can chain up to five input filters and four output filters together.
- If you do not configure and apply a filter, the interface uses the default group filter configuration.
- You cannot modify or delete a firewall filter while subscribers on the same logical interface are bound.

RELATED DOCUMENTATION

Understanding Dynamic Firewall Filters | 236

Fast Update Filters Overview | 344

Dynamically Attaching Statically Created Filters for Any Interface Type | 256

Dynamically Attaching Statically Created Filters for a Specific Interface Family Type | 255

Dynamically Attaching Filters Using RADIUS Variables | 307

Verifying and Managing Firewall Filter Configuration | 426

Basic Classic Filter Syntax

This section provides the basic classic filter CLI statement syntax. The first part of this syntax provides the CLI statements to associate an input and output filter with a dynamic profile. The second part of this syntax represents the configured input and output filters applied to the dynamic profile. When a DHCP event occurs, the dynamic profile applies the specified filters to the DHCP client interface on the router.

```
[edit]
dynamic-profiles [profile-name] {
    interfaces {
        [$junos-interface-ifd-name] {
            unit [$junos-underlying-interface-unit] {
                family  family] {
                    filter {
                        input {
                            [filter-name];
                            precedence [precedence];
                        }
                        output {
                            [filter-name];
                            precedence [precedence];
                        }
                    }
                }
            }
        }
    }
[edit]
firewall {
    family [family] {
        filter [filter-name] {
            [desired filter configuration]
        }
        filter [filter-name] {
            [desired filter configuration]
        }
    }
}
```

RELATED DOCUMENTATION

Dynamically Attaching Statically Created Filters for a Specific Interface Family Type | 255

Understanding Dynamic Firewall Filters | 236

Examples: Configuring Static Filters

This topic provides some static filter configuration examples.

```
firewall {
    policer p1 {
        if-exceeding {
            bandwidth-limit 5m;
            burst-size-limit 10m;
        }
        then discard;
    }
    family inet {
        filter dfwd {
            interface-specific;
            term 1 {
                from {
                    source-address {
                        192.51.100.10/24;
                    }
                }
                then {
                    count c1;
                    next term;
                }
            }
            term 2 {
                from {
                    source-address {
                        192.51.100.20/24;
                    }
                }
                then count c2;
            }
            term 3 {
```

```
                then accept;
            }
        }
        filter dfwd1 {
            interface-specific;
            term 1 {
                from {
                    address {
                        192.51.100.10/24;
                    }
                }
                then {
                    discard;
                }
            }
        }
        filter tos {
            interface-specific;
            term 1 {
                from {
                    precedence priority;
                }
                then forwarding-class assured-forwarding;
            }
            term 2 {
                then {
                    log;
                    accept;
                }
            }
        }
        filter dfwd2 {
            interface-specific;
            term 1 {
                from {
                    forwarding-class best-effort;
                }
                then {
                    sample;
                    forwarding-class expedited-forwarding;
                }
            }
            term 2 {
```

```
            then accept;
        }
    }
    filter nodhcp {
        term dhcpdiscover {
            from {
                protocol udp;
                source-port 68;
                destination-port 67;
            }
            then {
                discard;
            }
        }
        term others {
            then accept;
        }
    }
    filter p1 {
        interface-specific;
        term 1 {
            from {
                precedence priority;
            }
            then {
                policer p1;
                log;
            }
        }
        term 2 {
            then accept;
        }
    }
    filter dscp {
        interface-specific;
        term 1 {
            from {
                dscp af11;
            }
            then log;
        }
        term 2 {
            then accept;
```

```
            }
        }
        filter tcm {
            interface-specific;
            term 1 {
                from {
                    dscp af11;
                }
                then policer p1;
            }
            term 2 {
                then accept;
            }
        }
    }
    traceoptions {
        flag dynamic;
    }
}
```

RELATED DOCUMENTATION

Dynamically Attaching Statically Created Filters for Any Interface Type | 256

Dynamically Attaching Statically Created Filters for a Specific Interface Family Type | 255

CHAPTER 17

Streamlining Processing of Chains of Static Filters

IN THIS CHAPTER

- Configuring Firewall Filter Bypass | 248
- Example: Bypassing Firewall Filters | 249

Configuring Firewall Filter Bypass

You can streamline the filter process, decrease the amount of packet handling for each filter in a chain, and effectively bypass unnecessary filters by using the `service-filter-hit` match/action combination at the [`edit firewall family` *`family-name`* `filter` *`filter-name`* `term` *`term-name`*] hierarchy level.

To bypass firewall filters using the `service-filter-hit` match/action combination, you configure the `service-filter-hit` action in at least one filter in the chain and configure `service-filter-hit` match condition in any subsequent filters that you want to bypass. All packets must pass through each filter in a chain. However, after the `service-filter-hit` flag is set in a packet, the packet "bypasses" any subsequent filters that contain the `service-filter-hit` match condition and more efficiently passes (accepts) marked packets and accelerating the filter process.

NOTE: When using the `service-filter-hit` match/action combination, the order in which the filters are applied is important. You can ensure the order in which the filters are processed by specifying a filter precedence value for the interface. See "Defining Dynamic Filter Processing Order" on page 237 for more information about dynamic filter processing.

To bypass filter processing:

1. Specify the `service-filter-hit` action for any filters in a filter chain.

```
[edit firewall family inet filter video term 1]
user@host# set then service-filter-hit
```

When the match conditions for the filter are met, the `service-filter-hit` action is set to indicate to subsequent filters that further processing is unnecessary.

2. Specify the `service-filter-hit` match condition in any filters with a lower precedence (that is, a higher `precedence` statement value) that you want to detect `service-filter-hit` actions applied from previous filters in the chain.

```
[edit firewall family inet filter data term 1]
user@host# set from service-filter-hit
```

3. Configure the filter to pass (accept) any packet that has a `service-filter-hit` action applied from any previous filters.

```
[edit firewall family inet filter data term 1]
user@host# set then accept
```

RELATED DOCUMENTATION

- Classic Filters Overview | 239
- Defining Dynamic Filter Processing Order | 237
- Example: Bypassing Firewall Filters | 249

Example: Bypassing Firewall Filters

IN THIS SECTION

- Before You Begin | 250
- Filter Bypass Overview | 250
- Configuring Filter Bypass | 250

This example describes how to configure multiple filters using the `service-filter-hit` match/action combination and contains the following sections:

Before You Begin

When using the `service-filter-hit` match/action combination, keep the following in mind:

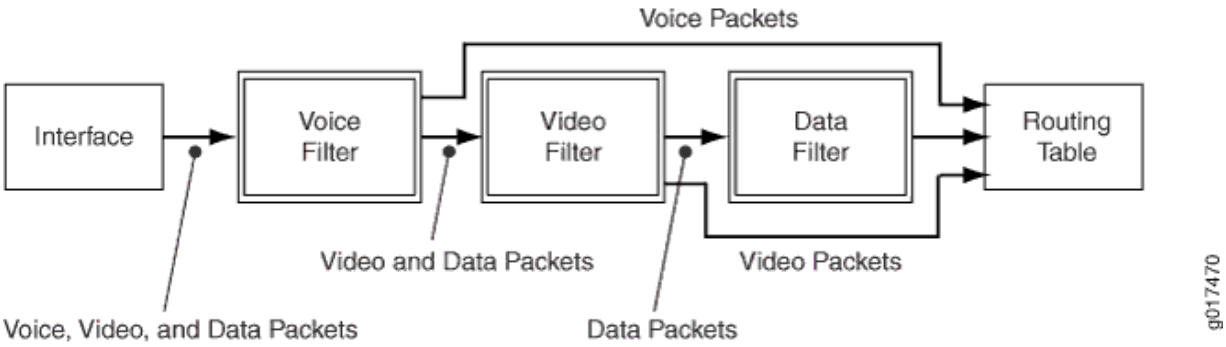
- The order in which the filters are applied is important. You can ensure the order in which the filters are processed by specifying a filter precedence value for the interface. See "Defining Dynamic Filter Processing Order" on page 237 for more information about dynamic filter processing and how to use the `precedence` statement.

Filter Bypass Overview

Packets must pass through each filter in a chain. However, if you create a chain of filters to process different types of packets (for example, voice, video, and data packets), you can streamline the filter process, decreasing the amount of packet handling for each filter in the chain, effectively bypassing unnecessary filters, by using the `service-filter-hit` match/action combination at the [`edit firewall family` *`family-name`* `filter` *`filter-name`* `term` *`term-name`*] hierarchy level.

Figure 5 on page 250 shows the logical processing flow through a chain of three filters (voice, video, and data) where only processing for a specific data type is desired. This configuration example shows an ingress filter flow. Though subsequent ingress filters in a chain can detect whether the `service-filter-hit` action is set, egress filters do not. To bypass egress filters, you must also configure the `service-filter-hit` match/action combination on those filters.

Figure 5: Logical Flow Example for Filter Bypass Processing



Configuring Filter Bypass

IN THIS SECTION

- CLI Quick Configuration | 251
- Configuring the Voice Filter | 251

- Configuring the Video Filter | 252
- Configuring the Data Filter | 252
- Results | 253

CLI Quick Configuration

To quickly configure this example:

```
[edit]
set firewall filter voice term T1 from address 203.0.113.11/32
set firewall filter voice term T1 from source-port 5004-5005
set firewall filter voice term T1 then forwarding-class assured-forwarding service-filter-hit
accept
set firewall filter voice term default then accept
set firewall filter video term T1 from service-filter-hit
set firewall filter video term T1 then accept
set firewall filter video term T2 from source-address 203.0.113.100/32
set firewall filter video term T2 then policer video-policer service-filter-hit accept
set firewall filter video term default then accept
set firewall filter data term T1 from service-filter-hit
set firewall filter data term T1 then accept
set firewall filter data term T2 then policer data-policer service-filter-hit accept
```

Configuring the Voice Filter

Step-by-Step Procedure

To configure the voice filter for the logical flow in Figure 5 on page 250:

1. Configure the filter to apply the assured forwarding class and set the `service-filter-hit` action for traffic from a specific address and port range (over which voice traffic is expected).

```
[edit]
set firewall filter voice term T1 from address 203.0.113.11/32
set firewall filter voice term T1 from source-port 5004-5005
```

```
set firewall filter voice term T1 then forwarding-class assured-forwarding service-filter-hit
accept
```

2. Configure the filter default action to pass (accept) packet traffic from any other address or port range.

```
[edit]
set firewall filter voice term default then accept
```

Configuring the Video Filter

Step-by-Step Procedure

To configure the video filter for the logical flow in Figure 5 on page 250:

1. Configure the filter to pass (accept) incoming packets that are tagged by the `service-filter-hit` action.

```
[edit]
set firewall filter video term T1 from service-filter-hit
set firewall filter video term T1 then accept
```

2. Configure the filter to apply a video policer and set the `service-filter-hit` action for traffic from a specific address (over which video traffic is expected).

```
[edit]
set firewall filter video term T2 from source-address 203.0.113.100/32
set firewall filter video term T2 then policer video-policer service-filter-hit accept
```

3. Configure the filter default action to pass (accept) packet traffic from any other address or port range.

```
[edit]
set firewall filter video term default then accept
```

Configuring the Data Filter

Step-by-Step Procedure

To configure the data filter for the logical flow in Figure 5 on page 250:

1. Configure the filter to pass (accept) incoming packets that are tagged by the `service-filter-hit` action.

```
[edit]
set firewall filter data term T1 from service-filter-hit
set firewall filter data term T1 then accept
```

2. Configure the filter to apply a data policer and set the `service-filter-hit` action for traffic from a specific address (over which video traffic is expected).

```
[edit]
set firewall filter data term T2 then policer data-policer service-filter-hit accept
```

Results

Display the results of the configuration:

```
[edit firewall]
user@host# show
filter voice {
    term T1 {
        from {
            address {
                203.0.113.11/32;
            }
            source-port 5004-5005;
        }
        then {
            forwarding-class assured-forwarding;
            service-filter-hit;
            accept;
        }
    }
    term default {
        then accept;
    }
}
filter video {
    term T1 {
        from {
            service-filter-hit;
```

```
        }
        then accept;
    }
    term T2 {
        from {
            source-address {
                203.0.113.100/32;
            }
        }
        then {
            policer video_policer;
            service-filter-hit;
            accept;
        }
    }
    term default {
        then accept;
    }
}
filter data {
    term T1 {
        from {
            service-filter-hit;
        }
        then accept;
    }
    term T2 {
        then {
            policer data_policer;
            service-filter-hit;
            accept;
        }
    }
}
```

RELATED DOCUMENTATION

- Classic Filters Overview | 239
- Defining Dynamic Filter Processing Order | 237
- Configuring Firewall Filter Bypass | 248

CHAPTER 18

Dynamically Attaching Static or Fast Update Filters to an Interface

IN THIS CHAPTER

- Dynamically Attaching Statically Created Filters for a Specific Interface Family Type | 255
- Dynamically Attaching Statically Created Filters for Any Interface Type | 256

Dynamically Attaching Statically Created Filters for a Specific Interface Family Type

Before you can attach a statically created filter using a dynamic profile.

1. Create the filters you want to attach.

   See Firewall Filters Overview for information about classic firewall filters and how to create them.
   See "Configuring Fast Update Filters" on page 349 for information about creating fast update filters.

2. Create a basic dynamic profile.

   See *Configuring a Basic Dynamic Profile*.

You can dynamically attach statically created filters for either IPv4 (`inet`) or IPv6 (`inet6`) interface types. These filters apply only to interfaces of the specified type.

To dynamically attach statically created input and output filters:

1. Specify the unit family type you want to use when dynamically attaching the filters.
   a. For IPv4 interfaces, specify the `inet` unit family.

```
[edit dynamic-profiles access-profile interfaces ge-1/1/1 unit 1]
user@host# set family inet
```

b. For IPv6 interfaces, specify the `inet6` unit family.

```
[edit dynamic-profiles access-profile interfaces ge-1/1/1 unit 1]
user@host# set family inet6
```

2. Specify the input filter in the dynamic profile.

```
[edit dynamic-profiles access-profile interfaces ge-1/1/1 unit 1 family inet]
user@host# set filter input static-input-filter
```

3. Specify the output filter in the dynamic profile.

NOTE: The following example specifies an optional precedence value for the output filter.

```
[edit dynamic-profiles access-profile interfaces ge-1/1/1 unit 1 family inet]
user@host# set filter output static-output-filter precedence 50
```

RELATED DOCUMENTATION

- Classic Filters Overview | 239
- Fast Update Filters Overview | 344
- Dynamically Attaching Statically Created Filters for Any Interface Type | 256
- Dynamically Attaching Filters Using RADIUS Variables | 307
- *Using the junos-defaults Configuration Group*
- Firewall Filters Overview

Dynamically Attaching Statically Created Filters for Any Interface Type

Before you can attach a statically created filter using a dynamic profile.

1. Create the filters you want to attach.

   See Firewall Filters Overview for information about classic firewall filters and how to create them.
   See "Configuring Fast Update Filters" on page 349 for information about creating fast update filters.

2. Create a basic dynamic profile.

   See *Configuring a Basic Dynamic Profile*.

You can dynamically attach statically created filters for any interface type. These filters apply to any interfaces that are created using the dynamic profile.

NOTE: For an L2TP LNS on MX Series routers, you can attach firewall for static LNS sessions by configuring these at logical interfaces directly on the inline services device (`si-fpc/pic/port`). RADIUS-configured firewall attachments are not supported.

To dynamically attach statically created input and output filters for all interfaces created dynamically using the dynamic profile:

1. Access the dynamic profile, interface, and unit that you want to use when applying the static filters.

```
[edit]
user@host# edit dynamic-profiles access-profile interfaces ge-1/1/1 unit 1
```

2. Specify the input filter for the interface unit.

```
[edit dynamic-profiles access-profile interfaces ge-1/1/1 unit 1]
user@host# set filter input static-input-filter
```

3. Specify the output filter for the interface unit.

```
[edit dynamic-profiles access-profile interfaces ge-1/1/1 unit 1]
user@host# set filter output static-output-filter
```

RELATED DOCUMENTATION

- Classic Filters Overview | 239
- Fast Update Filters Overview | 344
- Dynamically Attaching Statically Created Filters for a Specific Interface Family Type | 255
- Dynamically Attaching Filters Using RADIUS Variables | 307
- *Using the junos-defaults Configuration Group*
- Firewall Filters Overview

CHAPTER 19

Configuring Filters That Are Created Dynamically

IN THIS CHAPTER

- Parameterized Filters Overview | 258
- Unique Identifiers for Firewall Variables | 259
- Configuring Unique Identifiers for Parameterized Filters | 262
- Sample Dynamic-Profile Configuration for Parameterized Filters | 263
- Dynamic Profile After UID Substitutions for Parameterized Filters | 266
- Multiple Parameterized Filters | 268
- Parameterized Filter Processing Overview | 268
- Parameterized Filters Configuration Considerations | 270
- Guidelines for Creating and Applying Parameterized Filters for Subscriber Interfaces | 271
- Parameterized Filter Match Conditions for IPv4 Traffic | 272
- Parameterized Filter Match Conditions for IPv6 Traffic | 281
- Parameterized Filter Nonterminating and Terminating Actions and Modifiers | 291
- Firewall Filter Match Conditions for Protocol-Independent Traffic in Dynamic Service Profiles | 300
- Firewall Filter Terminating and Nonterminating Actions for Protocol-Independent Traffic in Dynamic Service Profiles | 302
- Interface-Shared Filters Overview | 306
- Dynamically Attaching Filters Using RADIUS Variables | 307
- Example: Implementing a Filter for Households That Use ACI-Based VLANs | 309
- Example: Dynamic-Profile Parsing | 312
- Example: Firewall Dynamic Profile | 313
- Example: Configuring a Filter to Exclude DHCPv6 and ICMPv6 Control Traffic for LAC Subscriber | 314

Parameterized Filters Overview

Parameterized filters allow you to implement customized filters for each subscriber session. In parameterized filters, you use variables called unique identifiers (UIDs) to define your filter. When

services are activated for a subscriber, actual values are substituted for the variables and are used to create filters.

Parameterized filters are configured under a dynamic profile. You can configure a general baseline filter under a dynamic profile and then provide specific variables of that filter when a dynamic session is activated. These variables can include policing rates, destination addresses, ports, and other items.

To provide better scaling, the system analyzes a dynamic profile, and then determines whether the set of variables for one session is the same as for a previous session. If a matching filter already exists, the session creates an interface-specific filter copy of that filter template. If the filter does not already exist, the session reads the configuration and compiles a new filter. This filter is installed as a template with an interface-specific filter copy for the current session pointing to it.

RELATED DOCUMENTATION

- Parameterized Filters Configuration Considerations | 270
- Parameterized Filter Processing Overview | 268
- Unique Identifiers for Firewall Variables | 259
- Sample Dynamic-Profile Configuration for Parameterized Filters | 263
- Dynamic Profile After UID Substitutions for Parameterized Filters | 266
- Example: Dynamic-Profile Parsing | 312
- Parameterized Filter Nonterminating and Terminating Actions and Modifiers | 291
- Parameterized Filter Match Conditions for IPv4 Traffic | 272
- Parameterized Filter Match Conditions for IPv6 Traffic | 281
- Understanding Dynamic Firewall Filters | 236

Unique Identifiers for Firewall Variables

The system uses unique identifiers (UIDs) to aid with scaling. The UID enables the system to determine when configuration objects from multiple subscribers are identical and can be shared. In many situations, such as a filter definition, sharing a single filter among multiple subscribers instead of creating a new filter for every subscriber helps to conserve system resources.

Within a dynamic profile a UID is used to name a configuration object. The system assigns the value of the UID (the object's name) based upon all the variables contained within that configuration stanza along with the dynamic profile's name. The assigned UID value consists of the UID name combined with the string `_UID` and a unique number. For instance, the UID `$my-filter` might be given the value `my-filter_UID1022`.

You must first define a UID under the `variable` stanza using the option `uid`. The UID must be defined at the end, after all the variables that are assigned values externally.

```
dynamic-profile test-profile {
    variables {
        ... [other variables] ...
        [my-filter] {
            uid;
        }
    }
}
```

After a UID has been defined, it can then be used to name an object:

```
dynamic-profile test-profile {
    firewall {
        family inet {
            filter [$my-filter] {
                ... [filter definition that makes use of other variables] ...
            }
        }
    }
}
```

As previously described, the system assigns the value of `$my-filter` depending on the values of the variables used within that filter's definition.

The UID is also used in any other place that the object's name is used. For example, here is an interface stanza to use `$my-filter` as an input filter:

```
dynamic-profile [test-profile] {
    interfaces {
        [$junos-interface-ifd-name]" {
            unit [$junos-interface-unit] {
                family inet {
                    filter {
                        input [$my-filter];
                    }
                }
            }
        }
```

```
    }
}
```

You can define multiple configuration objects of the same type (that is, multiple filters) as long as each one uses its own, individual, UID. To ensure that the system selects the correct object when assigning a name, use the **uid-reference** variable.

When the uid-reference is used, it is effectively evaluated twice. First, the value of the uid-reference variable is retrieved. Second, that value is used as the name of a UID and that UID value is retrieved. A uid-reference with a value that is not the name of a UID is considered an error.

A uid-reference is defined similarly to any other variable:

```
dynamic-profile [test-profile] {
    variables {
        [my-filter-selector] {
            uid-reference;
        }
    }
}
```

A uid-reference is used wherever the name of the object is needed. One example is the name of the input filter in the following interface stanza:

```
dynamic-profile [test-profile] {
    interfaces {
        [$junos-interface-ifd-name] {
            unit [$junos-interface-unit] {
                family inet {
                    filter {
                        input [$my-filter-selector];
                    }
                }
            }
        }
    }
}
```

Consider the case where two parameterized filters are defined: **$my-filter-1** and **$my-filter-2**. The **$my-filter-selector** variable might be assigned the value **my-filter-1** or **my-filter-2**, depending upon which filter is appropriate.

RELATED DOCUMENTATION

Configuring Unique Identifiers for Parameterized Filters | 262

Parameterized Filter Processing Overview | 268

Parameterized Filters Configuration Considerations | 270

Configuring Unique Identifiers for Parameterized Filters

This topic discusses how to configure unique identifiers (UIDs) that can then be used in parameterized filters. The dynamic profile obtains and replaces data for these variables from an incoming client data packet.

To configure unique identifiers for parameterized filters in a dynamic profile:

1. Access the desired dynamic profile.

```
[edit]
user@host# edit dynamic-profiles Profile1
[edit dynamic-profiles Profile1]
```

2. Configure the UIDs.

   If the value for the variable UID comes from RADIUS, configure the variable as a UID reference.

```
[edit dynamic-profiles Profile1]
user@host# set variable policer1 uid
user@host# set variable policer2 uid
user@host# set variable filter1 uid
user@host# set variable filter2 uid
user@host# set variables in-filter uid-reference
```

Example of UIDs that can be used in parameterized filters:

```
dynamic profile {
    Profile1 {
        variables {
            policer1 uid;
            filter1 uid;
            policer2 uid;
            filter2 uid;
```

```
            in-filter {
                uid-reference;
            }
        }
    }
```

RELATED DOCUMENTATION

- Unique Identifiers for Firewall Variables | 259
- Parameterized Filters Overview | 258
- *Dynamic Variables Overview*
- *Junos OS Predefined Variables*

Sample Dynamic-Profile Configuration for Parameterized Filters

In the following sample configuration, the `my-svc-prof` profile provides two different filters: `my-filt-1gw` and `my-filt-2gw`. These filters match on either one or two gateway addresses and apply a policer for that traffic. The name of the filter to apply, the gateway addresses, and the bandwidth for the policer are passed into the service profile from the RADIUS service activation. The uid-reference type supports selection of a particular UID generated object out of multiple objects in the profile. The UID type indicates that a variable is used for UID generation.

```
dynamic-profile {
    [my-svc-prof] {
        variable {
            [my-in-filter] {
                mandatory;
                uid-reference;
            }
            gw1 {
                mandatory;
            }
            gw2 {
                mandatory;
            }
            bw {
                mandatory;
```

```
            }
            my-filt-1gw {
                uid;
            }
            my-filt-2gw {
                uid;
            }
            [my-policer] {
                uid;
            }
        }

        interfaces {
            [$junos-interface-ifd-name] {
                unit [$junos-underlying-interface-unit] {
                    family inet {
                        filter {
                            input [$my-in-filter];
                        }
                    }
                }
            }
        }

        firewall {
            policer [$my-policer] {
                if-exceeding {
                    bandwidth-limit $bw;
                    burst-size-limit 15000;
                }
                then discard;
            }
            family inet {
                filter [$my-filt-1gw] {
                    interface-specific;
                    term t0 {
                        from {
                            destination-address $gw1;
                        }
                        then {
                            policer [$my-policer];
                        }
                    }
```

```
                    term last {
                        then {
                            count drops;
                            discard;
                        }
                    }
                }
                filter [$my-file-2gw] {
                    interface-specific;
                    term t0 {
                        from {
                            destination-address {
                                $gw1;
                                $gw2;
                            }
                        }
                        then {
                            policer [$my-policer];
                        }
                    }
                    term last {
                        then {
                            count drops;
                            discard;
                        }
                    }
                }
            }
        }
    }
}
```

RELATED DOCUMENTATION

- Dynamic Profile After UID Substitutions for Parameterized Filters | 266
- Example: Dynamic-Profile Parsing | 312
- Parameterized Filters Overview | 258
- Parameterized Filter Processing Overview | 268

Dynamic Profile After UID Substitutions for Parameterized Filters

In the following example, the client session is created on the ge-1/0/0.7 interface and this service is activated:

```
my-svc-prof(my-filt-1gw, 198.51.100.239/32, 0, 5m)
```

A dynamic profile is created by the process. The UIDs assigned by the process are based on the parameters being passed in as well as the sessions previously created.

```
dynamic-profile {
    [my-svc-prof] {

        interfaces {
            ge-1/0/0 {
                unit 7 {
                    family inet {
                        filter {
                            input my-filt-1gw_UID1022;
                        }
                    }
                }
            }
        }

        firewall {
            policer my-policer_UID1005 {
                if-exceeding {
                    bandwidth-limit 5m;
                    burst-size-limit 15000;
                }
                then discard;
            }
            family inet {
                filter my-filt-1gw_UID1022 {
                    interface-specific;
                    term t0 {
                        from {
                            destination-address 198.51.100.239/32;
                        }
                        then {
```

```
                        policer my-policer_UID1005;
                    }
                }
                term last {
                    then {
                        count drops;
                        discard;
                    }
                }
            }
            filter my-filt-2gw_UID11018 {
                interface-specific;
                term t0 {
                    from {
                        destination-address {
                            198.51.100.239/32;
                            0;
                        }
                    }
                    then {
                        policer my-policer_UID1005;
                    }
                }
                term last {
                    then {
                        count drops;
                        discard;
                    }
                }
            }
        }
    }
}
```

RELATED DOCUMENTATION

- Sample Dynamic-Profile Configuration for Parameterized Filters | 263
- Example: Dynamic-Profile Parsing | 312
- Parameterized Filters Overview | 258

Parameterized Filter Processing Overview | 268

Multiple Parameterized Filters

Differing filter match conditions can be achieved by allowing the filter that is being attached to be selected by the unique-identifier--reference capabilities of parameterized filters. If a variable number of terms or varying match conditions are needed, multiple filters are defined. When the service is activated, that activation will select the particular filter that should be applied in the stanza specifying the interface, unit, family and input/output filter:

```
interfaces {
    ge-1/0/0 {
        unit 7 {
            family inet {
                filter {
                    input my-filt-1gw-uid1022;
                }
            }
        }
    }
}
```

RELATED DOCUMENTATION

Parameterized Filters Overview | 258

Parameterized Filters Configuration Considerations | 270

Parameterized Filter Processing Overview

When creating a parameterized filter, you first define the family address type (`inet`, `inet6`, or `any`) and then you define one or more terms that specify the filtering criteria and the action to take when a match occurs.

Each term, or rule, consists of the following components:

- Match conditions—Specifies values or fields that the packet must contain. You can define various match conditions, including:

  - IP source address field
  - IP destination address field
  - Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) source port field
  - IP protocol field
  - Internet Control Message Protocol (ICMP) packet type
  - TCP flags
  - interfaces
- Actions—Specifies what to do when a match condition occurs. Possible actions are to accept or discard a packet. In addition, packets can be counted to collect statistical information. If no action is specified for a term, the default action is to accept the packet.

The processing of parameterized filters is the same as classic filters. The order of the terms within a parameterized filter is important. Packets are tested against each term in the order in which the terms are listed in the *firewall filter* configuration. When a firewall filter contains multiple terms, the router takes a top-down approach and compares a packet against the first term in the firewall filter. If the packet matches the first term, the router executes the action defined by that term to either accept or reject the packet, and no other terms are evaluated. If the router does not find a match between the packet and first term, it then compares the packet to the next term in the firewall filter by using the same match process. If no match occurs between the packet and the second term, the router continues to compare the packet to each successive term defined in the firewall filter until a match is found. If a packet does not match any terms in a firewall filter, the default action is to discard the packet.

You can also specify a precedence (from 0 through 255) for input and output filters within a dynamic profile to force filter processing in a particular order. Setting a lower precedence value for a filter gives it a higher precedence within the dynamic profile. Filters with lower precedence values are applied to interfaces before filters with higher precedence values. A precedence of zero (the default) gives the filter the highest precedence. If no precedence is specified, the filter receives a precedence of zero (highest precedence). Filters with matching precedence (zero or otherwise) are applied in an unspecified order.

NOTE: Parameterized filters do not support outbound packets that are sourced from the routing engine.

RELATED DOCUMENTATION

Parameterized Filters Configuration Considerations | 270

Parameterized Filters Configuration Considerations

IN THIS SECTION

- Subscriber IP Address | 270
- Interaction with Static Configuration | 270
- Interface-Specific Dynamic Service Filters | 271
- Service Session Support | 271
- Filter Naming Conventions | 271

Keep the following considerations in mind when configuring parameterized filters.

Subscriber IP Address

In most deployment scenarios, the interface is based on the subscriber's IP address. Because subscribers may not be unique, they cannot be used in determining similar filters and policers. Do not use the **junos-subscriber-ip-address IP** address as a match candidate. Doing so causes unique filters per subscriber, which inhibits scaling.

Interaction with Static Configuration

Searching for a filter to attach takes place in the following order:

1. Static filter. For example, **firewall family inet filter my-filter**.
2. Fast update filter within the current dynamic profile. For example, **dynamic-profile [*profile-name*] firewall family inet fast-update-filter my-filter**.
3. Parameterized filter within the current dynamic profile. For example, **dynamic-profile [*profile-name*] firewall family inet filter**.

The following static configuration objects may be referenced by a parameterized filter. The search order is first in the static configuration and then in the current dynamic-profile:

- firewall policer
- firewall hierarchical-policer
- three-color policer

- policy-options prefix-list

If an object in the static configuration is being used by an active parameterized filter, you cannot delete that object from the configuration while the subscriber is logged in.

Interface-Specific Dynamic Service Filters

All dynamic service filters must be defined as interface-specific.

Service Session Support

Parameterized filters and policers are supported for service activations only, not client sessions.

Filter Naming Conventions

The base filter name is based on the interface and direction (ingress and egress) appended to it. With parameterized filters, the filter-naming process comes from the UID.

RELATED DOCUMENTATION

Understanding Dynamic Firewall Filters | 236

Verifying and Managing Firewall Filter Configuration | 426

Unique Identifiers for Firewall Variables | 259

Sample Dynamic-Profile Configuration for Parameterized Filters | 263

Example: Dynamic-Profile Parsing | 312

Parameterized Filter Processing Overview | 268

Guidelines for Creating and Applying Parameterized Filters for Subscriber Interfaces

You can configure dynamic or static firewall filters. When you use statically configured firewall filters, you then dynamically apply those filters to statically created interfaces using dynamic profiles. You can also use dynamic profiles to attach input and output filters through RADIUS.

When creating and applying filters, keep the following in mind:

- Dynamic application of only input and output filters is supported.
- The filters must be interface-specific.

- You can create family-specific `any`, `inet`, and `inet6` filters.
- You can create interface-specific filters at the **unit** level that apply to any family type (`any`, `inet`, or `inet6`) configured on the interface.
- You can add or remove filters of different family types with the same service activation or deactivation.
- You can remove one filter type without impacting the other type of filter. For example, you can remove IPv6 filters and leave the current IPv4 filters active.
- You can chain up to five input filters and four output filters together.
- If you do not configure and apply a filter, the interface uses the default group filter configuration.
- You cannot modify a *firewall filter* while subscribers on the same *logical interface* are bound.

RELATED DOCUMENTATION

Parameterized Filter Processing Overview | 268

Parameterized Filters Configuration Considerations | 270

Parameterized Filter Match Conditions for IPv4 Traffic

You can configure a parameterized filter with match conditions for Internet Protocol version 4 (IPv4) traffic (`family inet`).

NOTE: For MX Series routers with MPCs, you need to initialize certain new firewall filters by walking the corresponding SNMP MIB, for example, `show snmp mib walk` *`name`* `ascii`. This forces Junos to learn the filter counters and ensure that the filter statistics are displayed. This guidance applies to all enhanced mode firewall filters, filters with flexible conditions, and filters with certain terminating actions. See those topics, listed under Related Documentation, for details.

Table 23 on page 273 describes the *`match-conditions`* you can configure at the `[edit firewall family inet filter` *`filter-name`* `term` *`term-name`* `from]` hierarchy level.

Table 23: Parameterized Filter Match Conditions for IPv4 Traffic

Match Condition	Description
address *address* [except]	Match the IPv4 source or destination address field unless the except option is included. If the option is included, do not match the IPv4 source or destination address field.
destination-address *address* [except]	Match the IPv4 destination address field unless the except option is included. If the option is included, do not match the IPv4 destination address field. You cannot specify both the address and destination-address match conditions in the same term.
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, 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. In place of the numeric value, you can specify one of the following text synonyms (the port numbers are also listed): 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), 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), ldp (646), login (513), mobileip-agent (434), mobilip-mn (435), 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), smtp (25), snmp (161), snmptrap (162), snpp (444), socks (1080), ssh (22), sunrpc (111), syslog (514), tacacs (49), tacacs-ds (65), talk (517), telnet (23), tftp (69), timed (525), who (513), or xdmcp (177).
destination-port-except *number*	Do not match the UDP or TCP destination port field. For details, see the destination-port match condition.

Table 23: Parameterized Filter Match Conditions for IPv4 Traffic *(Continued)*

Match Condition	Description
`destination-prefix-list` *`prefix-list-name`* `[ except ]`	Match destination prefixes in the specified list unless the `except` option is included. If the option is included, do not match the destination prefixes in the specified list. Specify the name of a prefix list defined at the [`edit policy-options prefix-list` *`prefix-list-name`*] hierarchy level. (Junos OS Evolved) You can specify prefix-list at the [`edit policy-options prefix-list` *`prefix-list-name`*] and [`edit dynamic-profiles` *`name`* `policy-options prefix-list` *`prefix-list-name`*] hierarchy levels.

Table 23: Parameterized Filter Match Conditions for IPv4 Traffic *(Continued)*

Match Condition	Description

`dscp` *number*	Match the Differentiated Services code point (DSCP). The DiffServ protocol uses the type-of-service (ToS) byte in the IP header. The most significant 6 bits of this byte form the DSCP. For more information, see Understanding How Behavior Aggregate Classifiers Prioritize Trusted Traffic.
	
	Starting in Junos OS Release 13.3R7, support was added for filtering on Differentiated Services Code Point (DSCP) and forwarding class for Routing Engine sourced packets, including IS-IS packets encapsulated in generic routing encapsulation (GRE). Subsequently, when upgrading from a previous version of Junos OS where you have both a class of service (CoS) and firewall filter, and both include DSCP or forwarding class filter actions, the criteria in the firewall filter automatically takes precedence over the CoS settings. The same is true when creating new configurations; that is, where the same settings exist, the firewall filter takes precedence over the CoS, regardless of which was created first.
	
	You can specify a numeric value from `0` through `63`. To specify the value in hexadecimal form, include `0x` as a prefix. To specify the value in binary form, include `b` as a prefix.
	
	In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed):
	
	• RFC 3246, *An Expedited Forwarding PHB (Per-Hop Behavior)*, defines one code point: `ef` (46).
	
	• RFC 2597, *Assured Forwarding PHB Group*, defines 4 classes, with 3 drop precedences in each class, for a total of 12 code points:
	
	  • `af11` (10), `af12` (12), `af13` (14)
	
	  • `af21` (18), `af22` (20), `af23` (22)
	
	  • `af31` (26), `af32` (28), `af33` (30)
	
	  • `af41` (34), `af42` (36), `af43` (38)

`dscp-except` *number*	Do not match on the DSCP number. For more information, see the `dscp` match condition.

Table 23: Parameterized Filter Match Conditions for IPv4 Traffic *(Continued)*

Match Condition	Description
`forwarding-class` *class*	Match the forwarding class of the packet. Specify `assured-forwarding`, `best-effort`, `expedited-forwarding`, or `network-control`. For information about forwarding classes and router-internal output queues, see Understanding How Forwarding Classes Assign Classes to Output Queues.
`forwarding-class-except` *class*	Do not match the forwarding class of the packet. For details, see the `forwarding-class` match condition.
`icmp-code` *number*	Match the ICMP message code field. If you configure this match condition, we recommend that you also configure the `protocol icmp` match condition in the same term. If you configure this match condition, you must also configure the `icmp-type` *message-type* match condition in the same term. An ICMP message code provides more specific information than an ICMP message type, but the meaning of an ICMP message code is dependent on the associated ICMP message type. 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 23: Parameterized Filter Match Conditions for IPv4 Traffic *(Continued)*

Match Condition	Description
`icmp-code-except` *`message-code`*	Do not match the ICMP message code field. For details, see the `icmp-code` match condition.
`icmp-type` *`number`*	Match the ICMP message type field. If you configure this match condition, we recommend that you also configure the `protocol icmp` match condition in the same term. 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).
`icmp-type-except` *`message-type`*	Do not match the ICMP message type field. For details, see the `icmp-type` match condition.
`loss-priority` *`level`*	Match the packet loss priority (PLP) level. Specify a single level or multiple levels: `low`, `medium-low`, `medium-high`, or `high`. For IP traffic on MX Series routers with Enhanced II Flexible PIC Concentrators (FPCs), you must include the `tri-color` statement at the `[edit class-of-service]` hierarchy level to commit a PLP configuration with any of the four levels specified. If the `tri-color` statement is not enabled, you can only configure the `high` and `low` levels. This applies to all protocol families. For information about the `tri-color` statement, see Configuring and Applying Tricolor Marking Policers. For information about using behavior aggregate (BA) classifiers to set the PLP level of incoming packets, see Understanding How Behavior Aggregate Classifiers Prioritize Trusted Traffic.
`loss-priority-except` *`level`*	Do not match the PLP level. For details, see the `loss-priority` match condition.
`packet-length` *`bytes`*	Match the length of the received packet, in bytes. The length refers only to the IP packet, including the packet header, and does not include any Layer 2 encapsulation overhead.

Table 23: Parameterized Filter Match Conditions for IPv4 Traffic *(Continued)*

Match Condition	Description
`packet-length-except` *bytes*	Do not match the length of the received packet, in bytes. For details, see the `packet-length` match type.
`port` *number*	Match the UDP or TCP source or destination port field. If you configure this match condition, you cannot configure the `destination-port` match condition or the `source-port` match condition in the same term. If you configure this match condition, 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. In place of the numeric value, you can specify one of the text synonyms listed under the `destination-port` match condition.
`port-except` *number*	Do not match either the source or destination UDP or TCP port field. For details, see the `port` match condition.
`precedence` *ip-precedence-value*	Match the IP precedence field. In place of the numeric field value, you can specify one of the following text synonyms (the field values are also listed): `critical-ecp` (0xa0), `flash` (0x60), `flash-override` (0x80), `immediate` (0x40), `internet-control` (0xc0), `net-control` (0xe0), `priority` (0x20), or `routine` (0x00). You can specify precedence in hexadecimal, binary, or decimal form.
`precedence-except` *ip-precedence-value*	Do not match the IP precedence field. In place of the numeric field value, you can specify one of the following text synonyms (the field values are also listed): `critical-ecp` (0xa0), `flash` (0x60), `flash-override` (0x80), `immediate` (0x40), `internet-control` (0xc0), `net-control` (0xe0), `priority` (0x20), or `routine` (0x00). You can specify precedence in hexadecimal, binary, or decimal form.

Table 23: Parameterized Filter Match Conditions for IPv4 Traffic *(Continued)*

Match Condition	Description
`prefix-list` *`prefix-list-name`* `[ except ]`	Match the prefixes of the source or destination address fields to the prefixes in the specified list unless the `except` option is included. If the option is included, do not match the prefixes of the source or destination address fields to the prefixes in the specified list. The prefix list is defined at the `[edit policy-options prefix-list` *`prefix-list-name`*`]` hierarchy level. (Junos OS Evolved) You can specify prefix-list at the `[edit policy-options prefix-list` *`prefix-list-name`*`]` and `[edit dynamic-profiles` *`name`* `policy-options prefix-list` *`prefix-list-name`*`]` hierarchy levels.
`protocol` *`number`*	Match the IP protocol type field. In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed): `ah` (51), `dstopts` (60), `egp` (8), `esp` (50), `fragment` (44), `gre` (47), `hop-by-hop` (0), `icmp` (1), `icmp6` (58), `icmpv6` (58), `igmp` (2), `ipip` (4), `ipv6` (41), `ospf` (89), `pim` (103), `rsvp` (46), `sctp` (132), `tcp` (6), `udp` (17), or `vrrp` (112).
`protocol-except` *`number`*	Do not match the IP protocol type field. In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed): `ah` (51), `dstopts` (60), `egp` (8), `esp` (50), `fragment` (44), `gre` (47), `hop-by-hop` (0), `icmp` (1), `icmp6` (58), `icmpv6` (58), `igmp` (2), `ipip` (4), `ipv6` (41), `ospf` (89), `pim` (103), `rsvp` (46), `sctp` (132), `tcp` (6), `udp` (17), or `vrrp` (112).
`service-filter-hit`	Match a packet received from a filter where a `service-filter-hit` action was applied.
`source-address` *`address`* `[ except ]`	Match the IPv4 address of the source node sending the packet unless the `except` option is included. If the option is included, do not match the IPv4 address of the source node sending the packet. You cannot specify both the `address` and `source-address` match conditions in the same term.
`source-class` *`class-names`*	Match one or more specified source class names (sets of source prefixes grouped together and given a class name). For more information, see Firewall Filter Match Conditions Based on Address Classes.

Table 23: Parameterized Filter Match Conditions for IPv4 Traffic *(Continued)*

Match Condition	Description
`source-class-except` *class-names*	Do not match one or more specified source class names. For details, see the `source-class` match condition.
`source-port` *number*	Match the UDP or TCP source port field. You cannot specify the `port` and `source-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. In place of the numeric value, you can specify one of the text synonyms listed with the `destination-port` *number* match condition.
`source-port-except` *number*	Do not match the UDP or TCP source port field. For details, see the `source-port` match condition.
`source-prefix-list` *name* `[ except ]`	Match source prefixes in the specified list unless the `except` option is included. If the option is included, do not match the source prefixes in the specified list. Specify the name of a prefix list defined at the `[edit policy-options prefix-list` *prefix-list-name*] hierarchy level. (Junos OS Evolved) You can specify prefix-list at the `[edit policy-options prefix-list` *prefix-list-name*] and `[edit dynamic-profiles` *name* `policy-options prefix-list` *prefix-list-name*] hierarchy levels.
`ttl` *number*	Match the IPv4 time-to-live number. Specify a TTL value or a range of TTL values. For *number*, you can specify one or more values from `0` through `255`. This match condition is supported only on M120, M320, MX Series, and T Series routers.
`ttl-except` *number*	Do not match on the IPv4 TTL number. For details, see the `ttl` match condition.

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
13.3R7	Starting in Junos OS Release 13.3R7, support was added for filtering on Differentiated Services Code Point (DSCP) and forwarding class for Routing Engine sourced packets, including IS-IS packets encapsulated in generic routing encapsulation (GRE).

RELATED DOCUMENTATION

- Parameterized Filters Overview | 258
- Guidelines for Creating and Applying Parameterized Filters for Subscriber Interfaces | 271
- Parameterized Filter Nonterminating and Terminating Actions and Modifiers | 291

Parameterized Filter Match Conditions for IPv6 Traffic

You can configure a parameterized filter with match conditions for Internet Protocol version 6 (IPv6) traffic (`family inet6`).

NOTE: For MX Series routers with MPCs, you need to initialize certain new firewall filters by walking the corresponding SNMP MIB, for example, `show snmp mib walk` *`name`* `ascii`. This forces Junos to learn the filter counters and ensure that the filter statistics are displayed. This guidance applies to all enhanced mode firewall filters, filters with flexible conditions, and filters with certain terminating actions. See those topics, listed under Related Documentation, for details.

Table 24 on page 282 describes the match conditions you can configure at the `[edit firewall family inet6 filter` *`filter-name`* `term` *`term-name`* `from]` hierarchy level.

Table 24: Firewall Filter Match Conditions for IPv6 Traffic

Match Condition	Description
address *address* [except]	Match the IPv6 source or destination address field unless the except option is included. If the option is included, do not match the IPv6 source or destination address field.
destination-address *address* [except]	Match the IPv6 destination address field unless the except option is included. If the option is included, do not match the IPv6 destination address field. You cannot specify both the address and destination-address match conditions in the same term.
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, we recommend that you also configure the next-header udp or next-header tcp match condition in the same term to specify which protocol is being used on the port. In place of the numeric value, you can specify one of the following text synonyms (the port numbers are also listed): 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), 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), ldp (646), login (513), mobileip-agent (434), mobilip-mn (435), 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), smtp (25), snmp (161), snmptrap (162), snpp (444), socks (1080), ssh (22), sunrpc (111), syslog (514), tacacs (49), tacacs-ds (65), talk (517), telnet (23), tftp (69), timed (525), who (513), or xdmcp (177).
destination-port-except *number*	Do not match the UDP or TCP destination port field. For details, see the destination-port match condition.

Table 24: Firewall Filter Match Conditions for IPv6 Traffic *(Continued)*

Match Condition	Description
`destination-prefix-list` *`prefix-list-name`* `[ except ]`	Match the IPv6 destination prefix to the specified list unless the `except` option is included. If the option is included, do not match the IPv6 destination prefix to the specified list. The prefix list is defined at the [`edit policy-options prefix-list` *`prefix-list-name`*] hierarchy level. (Junos OS Evolved) You can specify prefix-list at the [`edit policy-options prefix-list` *`prefix-list-name`*] and [`edit dynamic-profiles` *`name`* `policy-options prefix-list` *`prefix-list-name`*] hierarchy levels.
`forwarding-class` *`class`*	Match the forwarding class of the packet. Specify `assured-forwarding`, `best-effort`, `expedited-forwarding`, or `network-control`. For information about forwarding classes and router-internal output queues, see Understanding How Forwarding Classes Assign Classes to Output Queues.
`forwarding-class-except` *`class`*	Do not match the forwarding class of the packet. For details, see the `forwarding-class` match condition.

Table 24: Firewall Filter Match Conditions for IPv6 Traffic *(Continued)*

Match Condition	Description
`icmp-code` *message-code*	Match the ICMP message code field. If you configure this match condition, we recommend that you also configure the `next-header icmp` or `next-header icmp6` match condition in the same term. If you configure this match condition, you must also configure the `icmp-type` *message-type* match condition in the same term. An ICMP message code provides more specific information than an ICMP message type, but the meaning of an ICMP message code is dependent on the associated ICMP message type. 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: `ip6-header-bad` (0), `unrecognized-next-header` (1), `unrecognized-option` (2) • time-exceeded: `ttl-eq-zero-during-reassembly` (1), `ttl-eq-zero-during-transit` (0) • destination-unreachable: `administratively-prohibited` (1), `address-unreachable` (3), `no-route-to-destination` (0), `port-unreachable` (4)
`icmp-code-except` *message-code*	Do not match the ICMP message code field. For details, see the `icmp-code` match condition.

Table 24: Firewall Filter Match Conditions for IPv6 Traffic *(Continued)*

Match Condition	Description
`icmp-type` *message-type*	Match the ICMP message type field. If you configure this match condition, we recommend that you also configure the `next-header icmp` or `next-header icmp6` match condition in the same term. In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed): `certificate-path-advertisement` (149), `certificate-path-solicitation` (148), `destination-unreachable` (1), `echo-reply` (129), `echo-request` (128), `home-agent-address-discovery-reply` (145), `home-agent-address-discovery-request` (144), `inverse-neighbor-discovery-advertisement` (142), `inverse-neighbor-discovery-solicitation` (141), `membership-query` (130), `membership-report` (131), `membership-termination` (132), `mobile-prefix-advertisement-reply` (147), `mobile-prefix-solicitation` (146), `neighbor-advertisement` (136), `neighbor-solicit` (135), `node-information-reply` (140), `node-information-request` (139), `packet-too-big` (2), `parameter-problem` (4), `private-experimentation-100` (100), `private-experimentation-101` (101), `private-experimentation-200` (200), `private-experimentation-201` (201), `redirect` (137), `router-advertisement` (134), `router-renumbering` (138), `router-solicit` (133), or `time-exceeded` (3). For `private-experimentation-201` (201), you can also specify a range of values within square brackets.
`icmp-type-except` *message-type*	Do not match the ICMP message type field. For details, see the `icmp-type` match condition.

Table 24: Firewall Filter Match Conditions for IPv6 Traffic *(Continued)*

Match Condition	Description
`loss-priority` *level*	Match the packet loss priority (PLP) level. Specify a single level or multiple levels: `low`, `medium-low`, `medium-high`, or `high`. Supported on M120 and M320 routers; M7i and M10i routers with the Enhanced CFEB (CFEB-E); and MX Series routers and EX Series switches. For IP traffic on M320, MX Series, T Series routers and EX Series switches with Enhanced II Flexible PIC Concentrators (FPCs), you must include the `tri-color` statement at the `[edit class-of-service]` hierarchy level to commit a PLP configuration with any of the four levels specified. If the `tri-color` statement is not enabled, you can only configure the `high` and `low` levels. This applies to all protocol families. For information about the `tri-color` statement, see Configuring and Applying Tricolor Marking Policers. For information about using behavior aggregate (BA) classifiers to set the PLP level of incoming packets, see Understanding How Forwarding Classes Assign Classes to Output Queues.
`loss-priority-except` *level*	Do not match the PLP level. For details, see the `loss-priority` match condition.

Table 24: Firewall Filter Match Conditions for IPv6 Traffic *(Continued)*

Match Condition	Description
`next-header` *header-type*	Match the first 8-bit Next Header field in the packet. Support for the `next-header` firewall match condition is available in Junos OS Release 13.3R6 and later. For IPv6, we recommend that you use the `payload-protocol` term rather than the `next-header` term when configuring a firewall filter with match conditions. Although either can be used, `payload-protocol` provides the more reliable match condition because it uses the actual payload protocol to find a match, whereas `next-header` simply takes whatever appears in the first header following the IPv6 header, which may or may not be the actual protocol. In addition, if `next-header` is used with IPv6, the accelerated filter block lookup process is bypassed and the standard filter used instead. Match the first 8-bit Next Header field in the packet. In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed): `ah` (51), `dstops` (60), `egp` (8), `esp` (50), `fragment` (44), `gre` (47), `hop-by-hop` (0), `icmp` (1), `icmp6` (58), `icmpv6` (58), `igmp` (2), `ipip` (4), `ipv6` (41), `mobility` (135), `no-next-header` (59), `ospf` (89), `pim` (103), `routing` (43), `rsvp` (46), `sctp` (132), `tcp` (6), `udp` (17), or `vrrp` (112). **NOTE**: `next-header icmp6` and `next-header icmpv6` match conditions perform the same function. `next-header icmp6` is the preferred option. `next-header icmpv6` is hidden in the Junos OS CLI.
`next-header-except` *header-type*	Do not match the 8-bit Next Header field that identifies the type of header between the IPv6 header and payload. For details, see the `next-header` match type.
`packet-length` *bytes*	Match the length of the received packet, in bytes. The length refers only to the IP packet, including the packet header, and does not include any Layer 2 encapsulation overhead.
`packet-length-except` *bytes*	Do not match the length of the received packet, in bytes. For details, see the `packet-length` match type.

Table 24: Firewall Filter Match Conditions for IPv6 Traffic *(Continued)*

Match Condition	Description
`port` *number*	Match the UDP or TCP source or destination port field. If you configure this match condition, you cannot configure the `destination-port` match condition or the `source-port` match condition in the same term. If you configure this match condition, we recommend that you also configure the `next-header udp` or `next-header tcp` match condition in the same term to specify which protocol is being used on the port. In place of the numeric value, you can specify one of the text synonyms listed under the `destination-port` match condition.
`port-except` *number*	Do not match the UDP or TCP source or destination port field. For details, see the `port` match condition.
`prefix-list` *prefix-list-name* `[ except ]`	Match the prefixes of the source or destination address fields to the prefixes in the specified list unless the `except` option is included. If the option is included, do not match the prefixes of the source or destination address fields to the prefixes in the specified list. The prefix list is defined at the `[edit policy-options prefix-list` *prefix-list-name*`]` hierarchy level. (Junos OS Evolved) You can specify prefix-list at the `[edit policy-options prefix-list` *prefix-list-name*`]` and `[edit dynamic-profiles` *name* `policy-options prefix-list` *prefix-list-name*`]` hierarchy levels.
`service-filter-hit`	Match a packet received from a filter where a `service-filter-hit` action was applied.
`source-address` *address* `[ except ]`	Match the IPv6 address of the source node sending the packet unless the `except` option is included. If the option is included, do not match the IPv6 address of the source node sending the packet. You cannot specify both the `address` and `source-address` match conditions in the same term.

Table 24: Firewall Filter Match Conditions for IPv6 Traffic *(Continued)*

Match Condition	Description
`source-class` *class-names*	Match one or more specified source class names (sets of source prefixes grouped together and given a class name). For more information, see Firewall Filter Match Conditions Based on Address Classes.
`source-class-except` *class-names*	Do not match one or more specified source class names. For details, see the `source-class` match condition.
`source-port` *number*	Match the UDP or TCP source port field. You cannot specify the `port` and `source-port` match conditions in the same term. If you configure this match condition, we recommend that you also configure the `next-header udp` or `next-header tcp` match condition in the same term to specify which protocol is being used on the port. In place of the numeric value, you can specify one of the text synonyms listed with the `destination-port` *number* match condition.
`source-port-except` *number*	Do not match the UDP or TCP source port field. For details, see the `source-port` match condition.
`source-prefix-list` *name* `[ except ]`	Match the IPv6 address prefix of the packet source field unless the `except` option is included. If the option is included, do not match the IPv6 address prefix of the packet source field. Specify a prefix list name defined at the `[edit policy-options prefix-list` *prefix-list-name*`]` hierarchy level. (Junos OS Evolved) You can specify prefix-list at the `[edit policy-options prefix-list` *prefix-list-name*`]` and `[edit dynamic-profiles` *name* `policy-options prefix-list` *prefix-list-name*`]` hierarchy levels. The maximum prefix length that you can use is /64.

Table 24: Firewall Filter Match Conditions for IPv6 Traffic *(Continued)*

Match Condition	Description

`traffic-class` *number*	Match the 8-bit field that specifies the class-of-service (CoS) priority of the packet.
	
	This field was previously used as the type-of-service (ToS) field in IPv4.
	
	You can specify a numeric value from `0` through `63`. To specify the value in hexadecimal form, include `0x` as a prefix. To specify the value in binary form, include `b` as a prefix.
	
	In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed):
	
	• RFC 3246, *An Expedited Forwarding PHB (Per-Hop Behavior)*, defines one code point: `ef` (46).
	
	• RFC 2597, *Assured Forwarding PHB Group*, defines 4 classes, with 3 drop precedences in each class, for a total of 12 code points:
	
	  • `af11` (10), `af12` (12), `af13` (14)
	
	  • `af21` (18), `af22` (20), `af23` (22)
	
	  • `af31` (26), `af32` (28), `af33` (30)
	
	  • `af41` (34), `af42` (36), `af43` (38)

`traffic-class-except` *number*	Do not match the 8-bit field that specifies the CoS priority of the packet. For details, see the `traffic-class` match description.

NOTE: If you specify an IPv6 address in a match condition (the `address`, `destination-address`, or `source-address` match conditions), use the syntax for text representations described in RFC 4291, *IP Version 6 Addressing Architecture*. For more information about IPv6 addresses, see IPv6 Overview and Supported IPv6 Standards.

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
13.3R6	Support for the `next-header` firewall match condition is available in Junos OS Release 13.3R6 and later.

RELATED DOCUMENTATION

- Guidelines for Configuring Firewall Filters
- Firewall Filter Terminating Actions
- Firewall Filter Nonterminating Actions
- Firewall Filter Match Conditions for IPv4 Traffic
- *enhanced-mode*
- Firewall Filter Flexible Match Conditions

Parameterized Filter Nonterminating and Terminating Actions and Modifiers

The nonterminating and terminating actions and modifiers for parameterized filters are a subset of those available for static firewall filters.

NOTE: You cannot configure the `next term` *nonterminating* action with a *terminating* action in the same filter term. However, you can configure the `next term` action with another *nonterminating* action in the same filter term.

Nonterminating actions carry with them an implicit accept action. In this context, *nonterminating* means that other actions can follow these actions, whereas no other actions can follow a *terminating* action.

Table 25 on page 292 describes the nonterminating actions and modifiers you can configure for a parameterized filter term.

Table 25: Nonterminating Actions for Parameterized Filters

Nonterminating Action	Description	Protocol Families
`count` *`counter-name`*	Count the packet in the named counter.	• `family any` • `family inet` • `family inet6`

Table 25: Nonterminating Actions for Parameterized Filters *(Continued)*

Nonterminating Action	Description	Protocol Families
`dscp` *value*	Set the IPv4 Differentiated Services code point (DSCP) bit. You can specify a numerical value from `0` through `63`. To specify the value in hexadecimal form, include `0x` as a prefix. To specify the value in binary form, include `b` as a prefix. The default DSCP value is best effort, that is, `be` or `0`. You can also specify one of the following text synonyms: • `af11`—Assured forwarding class 1, low drop precedence • `af12`—Assured forwarding class 1, medium drop precedence • `af13`—Assured forwarding class 1, high drop precedence • `af21`—Assured forwarding class 2, low drop precedence • `af22`—Assured forwarding class 2, medium drop precedence • `af23`—Assured forwarding class 2, high drop precedence • `af31`—Assured forwarding class 3, low drop precedence • `af32`—Assured forwarding class 3, medium drop precedence • `af33`—Assured forwarding class 3, high drop precedence • `af41`—Assured forwarding class 4, low drop precedence • `af42`—Assured forwarding class 4, medium drop precedence • `af43`—Assured forwarding class 4, high drop precedence • `be`—Best effort • `cs0`—Class selector 0 • `cs1`—Class selector 1 • `cs2`—Class selector 2 • `cs3`—Class selector 3	`family inet`

Table 25: Nonterminating Actions for Parameterized Filters *(Continued)*

Nonterminating Action	Description	Protocol Families
	• cs4—Class selector 4 • cs5—Class selector 5 • cs6—Class selector 6 • cs7—Class selector 7 • ef—Expedited forwarding	
forwarding-class *class-name*	Classify the packet to the named forwarding class: • assured-forwarding • best-effort • expedited-forwarding • network-control	• family any • family inet • family inet6
hierarchical-policer	Police the packet using the specified hierarchical policer.	• family any • family inet • family inet6
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 CLI. **NOTE**: The Layer 2 (L2) families log action is available only for MX Series routers with MPCs (MPC mode if the router has only MPCs, or mix mode if it has MPCs and DCPs). For MX Series routers with DPCs, the log action for L2 families is ignored if configured.	• family inet • family inet6

Table 25: Nonterminating Actions for Parameterized Filters *(Continued)*

Nonterminating Action	Description	Protocol Families
`loss-priority (high | medium-high | medium-low | low)`	Set the packet loss priority (PLP) level. You cannot also configure the `three-color-policer` nonterminating action for the same firewall filter term. These two nonterminating actions are mutually exclusive. For IP traffic on MX Series routers with Enhanced II Flexible PIC Concentrators (FPCs), you must include the `tri-color` statement at the `[edit class-of-service]` hierarchy level to commit a PLP configuration with any of the four levels specified. If the `tri-color` statement is not enabled, you can only configure the `high` and `low` levels. This applies to all protocol families. For information about the `tri-color` statement and using behavior aggregate (BA) classifiers to set the PLP level of incoming packets, see Understanding How Behavior Aggregate Classifiers Prioritize Trusted Traffic.	• `family any` • `family inet` • `family inet6`
`next`	Proceed to the next filter term.	• `family any` • `family inet` • `family inet6`

`next-ip` *`ip-address`* `<routing-instance` *`routing-instance`*`>`	(MX Series) Direct packets to the specified destination IPv4 address. You can optionally specify a routing instance for the address. In the following example, the variables $IP-address and $RT-name would be defined in `[edit dynamic-profiles` *`service-profile-name`* `variables]`:
	
	`[edit dynamic-profiles` *`service-profile-name`* `firewall family inet filter $nextip]`
	`user@host# set term t1 then next-ip $IP-address routing-instance $RT-name`
	
	Supported starting in Junos OS Release 18.2R1.
		`family inet`

Table 25: Nonterminating Actions for Parameterized Filters *(Continued)*

Nonterminating Action	Description	Protocol Families

`next-ip6 ipv6-address <routing-instance routing-instance>`	(MX Series) Direct packets to the specified destination IPv6 address. You can optionally specify a routing instance for the address. In the following example, the variables $IPv6-address and $RT-name would be defined in `[edit dynamic-profiles service-profile-name variables]`
	
	`[edit dynamic-profiles service-profile-name firewall family inet filter $nextip6]`
	`user@host# set term t1 then next-ip6 $IPv6-address routing-instance $RT-name`
	
	Supported starting in Junos OS Release 18.2R1.
		`family inet6`

`policer policer-name`	Name of policer to use to rate-limit traffic.	• `family any` • `family inet` • `family inet6`
`port-mirror instance-name`	Port-mirror the packet based on the specified family. We recommend that you do not use both the `next-hop-group` and the `port-mirror` actions in the same firewall filter.	• `family any` • `family inet` • `family inet6`
`port-mirror-instance instance-name`	Port-mirror a packet for an instance. This action is supported only on the MX Series routers. We recommend that you do not use both the `next-hop-group` and the `port-mirror-instance` actions in the same firewall filter.	• `family any` • `family inet` • `family inet6`
`routing-instance routing-instance-name`	Direct packets to the specified routing instance.	• `family inet` • `family inet6`

Table 25: Nonterminating Actions for Parameterized Filters *(Continued)*

Nonterminating Action	Description	Protocol Families
`sample`	Sample the packet. **NOTE**: Junos OS does not sample packets originating from the router. If you configure a filter and apply it to the output side of an interface, then only the transit packets going through that interface are sampled. Packets that are sent from the Routing Engine to the Packet Forwarding Engine are not sampled.	• `family inet` • `family inet6`
`service-accounting`	Use the inline counting mechanism when capturing subscriber per-service statistics. Count the packet for service accounting. The count is applied to a specific named counter (`__junos-dyn-service-counter`) that RADIUS can obtain. The `service-accounting` and `service-accounting-deferred` keywords are mutually exclusive, both per-term and per-filter.	• `family any` • `family inet` • `family inet6`
`service-accounting-deferred`	Use the deferred counting mechanism when capturing subscriber per-service statistics. The count is applied to a specific named counter (`__junos-dyn-service-counter`) that RADIUS can obtain. The `service-accounting` and `service-accounting-deferred` keywords are mutually exclusive, both per-term and per-filter.	• `family any` • `family inet` • `family inet6`
`service-filter-hit`	(Only if the `service-filter-hit` flag is marked by a previous filter in the current type of chained filters) Direct the packet to the next type of filters. Indicate to subsequent filters in the chain that the packet was already processed. This action, coupled with the `service-filter-hit` match condition in receiving filters, helps to streamline filter processing.	• `family any` • `family inet` • `family inet6`
`three-color-policer (single-rate | two-rate)` *`policer-name`*	Police the packet using the specified single-rate or two-rate three-color-policer. **NOTE**: You cannot also configure the `loss-priority` action for the same firewall filter term. These two actions are mutually exclusive.	• `family any` • `family inet` • `family inet6`

Table 25: Nonterminating Actions for Parameterized Filters *(Continued)*

Nonterminating Action	Description	Protocol Families
`traffic-class` *value*	Specify the traffic-class code point. You can specify a numerical value from `0` through `63`. To specify the value in hexadecimal form, include `0x` as a prefix. To specify the value in binary form, include `b` as a prefix. The default traffic-class value is best effort, that is, `be` or `0`. In place of the numeric value, you can specify one of the following text synonyms: • `af11`—Assured forwarding class 1, low drop precedence • `af12`—Assured forwarding class 1, medium drop precedence • `af13`—Assured forwarding class 1, high drop precedence • `af21`—Assured forwarding class 2, low drop precedence • `af22`—Assured forwarding class 2, medium drop precedence • `af23`—Assured forwarding class 2, high drop precedence • `af31`—Assured forwarding class 3, low drop precedence • `af32`—Assured forwarding class 3, medium drop precedence • `af33`—Assured forwarding class 3, high drop precedence • `af41`—Assured forwarding class 4, low drop precedence • `af42`—Assured forwarding class 4, medium drop precedence • `af43`—Assured forwarding class 4, high drop precedence • `be`—Best effort • `cs0`—Class selector 0 • `cs1`—Class selector 1 • `cs2`—Class selector 2 • `cs3`—Class selector 3	`family inet6`

Table 25: Nonterminating Actions for Parameterized Filters *(Continued)*

Nonterminating Action	Description	Protocol Families
	• `cs4`—Class selector 4 • `cs5`—Class selector 5 • `cs6`—Class selector 6 • `cs7`—Class selector 7 • `ef`—Expedited forwarding	

Table 26 on page 299 describes the terminating actions and modifiers you can configure for a parameterized filter term.

Table 26: Terminating Actions for Parameterized Filters

Terminating Action	Description	Protocol Families
accept	Accept the packet.	• `family any` • `family inet` • `family inet6`
`discard`	Discard a packet silently, without sending an Internet Control Message Protocol (ICMP) message. Discarded packets are available for logging and sampling.	• `family any` • `family inet` • `family inet6`

Table 26: Terminating Actions for Parameterized Filters *(Continued)*

Terminating Action	Description	Protocol Families
`reject` *`message-type`*	Reject the packet and return an ICMPv4 or ICMPv6 message: • If no *`message-type`* is specified, a `destination unreachable` message is returned by default. • If `tcp-reset` is specified as the *`message-type`*, `tcp-reset` is returned only if the packet is a TCP packet. Otherwise, the `administratively-prohibited` message, which has a value of 13, is returned. • If any other *`message-type`* is specified, that message is returned. **NOTE**: Rejected packets can be sampled or logged if you configure the `sample` or `syslog` action. The *`message-type`* can be one of the following values: `address-unreachable`, `administratively-prohibited`, `bad-host-tos`, `bad-network-tos`, `beyond-scope`, `fragmentation-needed`, `host-prohibited`, `host-unknown`, `host-unreachable`, `network-prohibited`, `network-unknown`, `network-unreachable`, `no-route`, `port-unreachable`, `precedence-cutoff`, `precedence-violation`, `protocol-unreachable`, `source-host-isolated`, `source-route-failed`, or `tcp-reset`.	• `family inet` • `family inet6`

RELATED DOCUMENTATION

- Parameterized Filters Overview | 258
- Guidelines for Creating and Applying Parameterized Filters for Subscriber Interfaces | 271
- Parameterized Filter Match Conditions for IPv4 Traffic | 272
- Parameterized Filter Match Conditions for IPv6 Traffic | 281
- Understanding Filter-Based Forwarding to a Specific Outgoing Interface or Destination IP Address

Firewall Filter Match Conditions for Protocol-Independent Traffic in Dynamic Service Profiles

You configure firewall filter match conditions to determine which packets are filtered. Starting in Junos OS Release 16.1, you can configure match conditions that are supported for protocol-independent

traffic—that is, configured under `family any`—for filters in dynamic service profiles. Table 27 on page 301 describes these match conditions.

NOTE: Protocol-independent firewall filters in dynamic service profiles are supported only on MX Series routers with MPCs.

Table 27: Firewall Filter Match Conditions for Protocol-Independent Traffic in Dynamic Service Profiles

Match Condition	Description
`forwarding-class` *class*	Match the forwarding class of the packet. Specify `assured-forwarding`, `best-effort`, `expedited-forwarding`, or `network-control`. For information about forwarding classes and router-internal output queues, see Understanding How Forwarding Classes Assign Classes to Output Queues.
`forwarding-class-except` *class*	Do not match on the forwarding class. For details, see the `forwarding-class` match condition.
`loss-priority` *level*	Match the packet loss priority (PLP) level. Specify a single level or multiple levels: `low`, `medium-low`, `medium-high`, or `high`. For information about the `tri-color` statement, see Configuring and Applying Tricolor Marking Policers. For information about using behavior aggregate (BA) classifiers to set the PLP level of incoming packets, see Understanding How Forwarding Classes Assign Classes to Output Queues.
`loss-priority-except` *level*	Do not match the PLP level. For details, see the `loss-priority` match condition.
`packet-length` *bytes*	Match the length of the received packet, in bytes. The length refers only to the IP packet, including the packet header, and does not include any Layer 2 encapsulation overhead.
`packet-length-except` *bytes*	Do not match on the received packet length, in bytes. For details, see the `packet-length` match type.

Table 27: Firewall Filter Match Conditions for Protocol-Independent Traffic in Dynamic Service Profiles *(Continued)*

Match Condition	Description
`service-filter-hit`	(Only if the `service-filter-hit` flag is marked by a previous filter in the current type of chained filters) Direct the packet to the next type of filters. Indicate to subsequent filters in the chain that the packet was already processed. This match option, coupled with the `service-filter-hit` nonterminating action, helps to streamline filter processing.

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
16.1	Starting in Junos OS Release 16.1, you can configure match conditions that are supported for protocol-independent traffic—that is, configured under `family any`—for filters in dynamic service profiles.

RELATED DOCUMENTATION

- Guidelines for Configuring Firewall Filters
- Firewall Filter Terminating and Nonterminating Actions for Protocol-Independent Traffic in Dynamic Service Profiles | **302**
- Firewall Filter Match Conditions for IPv4 Traffic
- Firewall Filter Match Conditions for IPv6 Traffic

Firewall Filter Terminating and Nonterminating Actions for Protocol-Independent Traffic in Dynamic Service Profiles

Firewall filters in dynamic service profiles support a set of terminating actions that halt all evaluation of a firewall filter for a specific packet. The router performs the specified action, and no additional terms are examined. Table 28 on page 303 describes the terminating actions conditions that are supported for protocol-independent traffic—that is, configured under `family any`—for filters in dynamic service profiles.

NOTE: You cannot configure the **next** action with a *terminating* action in the same filter term. However, you can configure the **next** action with another *nonterminating* action in the same filter term.

NOTE: Protocol-independent firewall filters in dynamic service profiles are supported only on MX Series routers with MPCs.

Table 28: Terminating Actions for Firewall Filters for Protocol-Independent Traffic in Dynamic Service Profiles

Terminating Action	Description
accept	Accept the packet.
`discard`	Discard a packet silently, without sending an Internet Control Message Protocol (ICMP) message. Discarded packets are available for logging and sampling.

Firewall filters in dynamic service profiles also support a set of nonterminating actions that are performed for a specific packet before the packet is passed to any subsequent actions in the term. Table 28 on page 303 describes the terminating actions conditions that are supported for protocol-independent traffic—that is, configured under `family any`—for filters in dynamic service profiles.

Table 29: Nonterminating Actions for Firewall Filters for Protocol-Independent Traffic in Dynamic Service Profiles

Nonterminating Action	Description
`count` *`counter-name`*	Count the packet in the named counter.
`force-premium`	By default, a hierarchical policer processes the traffic it receives according to the traffic's forwarding class. Premium, expedited-forwarding traffic, has priority for bandwidth over aggregate, best-effort traffic. The `force-premium` filter ensures that traffic matching the term is treated as premium traffic by a subsequent hierarchical policer, regardless of its forwarding class. This traffic is given preference over any aggregate traffic received by that policer. **NOTE**: The `force-premium` filter option is supported only on MPCs.

Table 29: Nonterminating Actions for Firewall Filters for Protocol-Independent Traffic in Dynamic Service Profiles *(Continued)*

Nonterminating Action	Description
`forwarding-class` *`class-name`*	Classify the packet to the named forwarding class: • *forwarding-class-name* • `assured-forwarding` • `best-effort` • `expedited-forwarding` • `network-control` **NOTE**: This action is supported on ingress only.
`hierarchical-policer`	Police the packet using the specified hierarchical policer.
`loss-priority (high | medium-high | medium-low | low)`	Set the packet loss priority (PLP) level. You cannot also configure the `three-color-policer` nonterminating action for the same firewall filter term. These two nonterminating actions are mutually exclusive. You must include the `tri-color` statement at the `[edit class-of-service]` hierarchy level to commit a PLP configuration with any of the four levels specified. If the `tri-color` statement is not enabled, you can configure only the `high` and `low` levels. This applies to all protocol families. For information about the `tri-color` statement, see Configuring and Applying Tricolor Marking Policers. For information about using behavior aggregate (BA) classifiers to set the PLP level of incoming packets, see Understanding How Forwarding Classes Assign Classes to Output Queues. For information about the `tri-color` statement and using behavior aggregate (BA) classifiers to set the PLP level of incoming packets, see Understanding How Behavior Aggregate Classifiers Prioritize Trusted Traffic. **NOTE**: This action is supported on ingress only.
`next`	Proceed to the next filter term.

Table 29: Nonterminating Actions for Firewall Filters for Protocol-Independent Traffic in Dynamic Service Profiles *(Continued)*

Nonterminating Action	Description
`policer` *`policer-name`*	Name of policer to use to rate-limit traffic.
`port-mirror` *`instance-name`*	Port-mirror the packet based on the specified family. **NOTE**: This action is supported on ingress only.
`service-accounting`	Use the inline counting mechanism when capturing subscriber per-service statistics. Count the packet for service accounting. The count is applied to a specific named counter (`__junos-dyn-service-counter`) that RADIUS can obtain. The `service-accounting` and `service-accounting-deferred` keywords are mutually exclusive, both per-term and per-filter. **NOTE**: This action is not supported on T4000 Type 5 FPCs and PTX Series Packet Transport Routers.
`service-accounting-deferred`	Use the deferred counting mechanism when capturing subscriber per-service statistics. The count is applied to a specific named counter (`__junos-dyn-service-counter`) that RADIUS can obtain. The `service-accounting` and `service-accounting-deferred` keywords are mutually exclusive, both per-term and per-filter. **NOTE**: This action is not supported on T4000 Type 5 FPCs and PTX Series Packet Transport Routers.
`service-filter-hit`	(Only if the `service-filter-hit` flag is marked by a previous filter in the current type of chained filters) Direct the packet to the next type of filters. Indicate to subsequent filters in the chain that the packet was already processed. This action, coupled with the `service-filter-hit` match condition in receiving filters, helps to streamline filter processing. **NOTE**: This action is not supported on T4000 Type 5 FPCs and PTX Series Packet Transport Routers.

Table 29: Nonterminating Actions for Firewall Filters for Protocol-Independent Traffic in Dynamic Service Profiles *(Continued)*

Nonterminating Action	Description
`three-color-policer (single-rate | two-rate)` *policer-name*	Police the packet using the specified single-rate or two-rate three-color-policer. **NOTE**: You cannot also configure the `loss-priority` action for the same firewall filter term. These two actions are mutually exclusive.

RELATED DOCUMENTATION

- Guidelines for Configuring Firewall Filters
- Firewall Filter Nonterminating Actions
- Firewall Filter Terminating Actions
- Firewall Filter Match Conditions for Protocol-Independent Traffic in Dynamic Service Profiles | 300

Interface-Shared Filters Overview

Interface-shared filters can be defined statically or dynamically, but can only be applied using dynamic profiles, and are supported for both client and service sessions. The same interface-shared instance can be attached to multiple interfaces only if these interfaces reference the same interface-shared filter name and have the same shared name.

The shared name can be taken from either the `$junos-interface-set-name` variable or the `$junos-svlan-interface-set-name` variable, where the values of the variables are provided by the related client session or service session. For example, if the `$junos-interface-set-name` variable is defined as the shared name, the same interface-shared filter instance is attached to all logical interfaces that belong to the interface set defined by the variable of that session. Similarly, if `$junos-svlan-interface-set-name` is defined for the shared name, all logical interfaces that belong to the VLAN interfaces set defined by the session's variable share the same interface-shared instance.

With VLAN subscriber interfaces that use the agent-circuit-identifier information, many subscribers share the same underlying *logical interface*. Because some of these subscribers are related to each other as part of the same household, you must apply an interface-shared filter to the subscriber logical interfaces that make up the household to be able to filter and police these related subscribers at a

household level. All interfaces that share the same interface-shared filter instance share the same set of counters and policer actions.

The base filter name of a parameterized filter is assigned depending upon the profile name and the contents of the filter definition. Therefore, when an interface-shared filter is used with parameterized filters, all service sessions that want to share the same instance of an interface-shared filter must have the exact same parameterized filter and profile. A service session uses a different instance of the interface-shared filter if either the parameterized filter or the profile is different.

RELATED DOCUMENTATION

Example: Implementing a Filter for Households That Use ACI-Based VLANs | 309

Dynamically Attaching Filters Using RADIUS Variables

You can attach filters to static interfaces by using dynamic profiles. By specifying a variable for the input and output filters, the dynamic profile uses RADIUS VSA attributes for ingress and egress policy.

RADIUS VSA	Attribute Name	Variable
26–10	Ingress-Policy-Name	$junos-input-filter
26–11	Egress-Policy-Name	$junos-output-filter
26–106	IPv6-Ingress-Policy-Name	$junos-input-ipv6-filter
26–107	IPv6-Egress-Policy-Name	$junos-output-ipv6-filter
26–191	Input-Interface-Filter	$junos-input-interface-filter
26–192	Output-Interface-Filter	$junos-output-interface-filter

To dynamically attach IPv4 input and output filters using RADIUS:

1. Specify the dynamic profile you want to attach, the interface, the logical unit number, and family `inet`.

```
[edit]
user@host# edit dynamic-profiles myProfile interface ge-1/1/1 unit 1 family inet
```

2. Specify the IPv4 input filter variable in the dynamic profile.

```
[edit dynamic-profiles myProfile interfaces ge-1/1/1 unit 1 family inet]
user@host# set filter input $junos-input-filter
```

3. Specify the IPv4 output filter variable in the dynamic profile.

```
[edit dynamic-profiles myProfile interfaces ge-1/1/1 unit 1 family inet]
user@host# set filter output $junos-output-filter
```

To dynamically attach IPv6 input and output filters using RADIUS:

1. Specify the dynamic profile you want to attach, the interface, the logical unit number, and family `inet6`.

```
[edit]
user@host# edit dynamic-profiles myProfile interface ge-1/1/1 unit 1 family inet6
```

2. Specify the IPv6 input filter variable in the dynamic profile.

```
[edit dynamic-profiles myProfile interfaces ge-1/1/1 unit 1 family inet6]
user@host# set filter input $junos-input-ipv6-filter
```

3. Specify the IPv6 output filter variable in the dynamic profile.

```
[edit dynamic-profiles myProfile interfaces ge-1/1/1 unit 1 family inet6]
user@host# set filter output $junos-output-ipv6-filter
```

To dynamically attach input and output filters to any interface independent of protocol using RADIUS:

1. Specify the dynamic profile you want to attach, the interface, and the logical unit number.

```
[edit]
user@host# edit dynamic-profiles myProfile interfaces ge-1/1/1 unit 1
```

2. Specify the input filter variable that applies to all families configured for the logical interface.

```
[edit dynamic-profiles myProfile interfaces ge-1/1/1 unit 1
user@host# set filter input $junos-input-interface-filter
```

3. Specify the output filter variable that applies to all families configured for the logical interface.

```
[edit dynamic-profiles myProfile interfaces ge-1/1/1 unit 1]
user@host# set filter output $junos-output-interface-filter
```

RELATED DOCUMENTATION

- Classic Filters Overview | 239
- Dynamically Attaching Statically Created Filters for Any Interface Type | 256
- Dynamically Attaching Statically Created Filters for a Specific Interface Family Type | 255
- *Junos OS Predefined Variables*
- *Using the junos-defaults Configuration Group*
- Firewall Filters Overview

Example: Implementing a Filter for Households That Use ACI-Based VLANs

In the following example using an interface-shared filter, you configure a dynamic profile that is used to implement agent-circuit-identifier VLAN household filtering. If `$junos-input-filter` is FILTER1 and `$junos-interface-set-name` is ACI1, then a filter with the name FILTER1-ACI1-in is created and attached to the demux0 unit. When a subsequent login from the same household occurs, it is in the same VLAN. If `$junos-input-filter` is also FILTER1, the next demux0 interface also has the FILTER1-ACI1-in filter attached. A low value precedence was used with the interface-shared filter. If you want to have the

interface-shared filter applied first, give a higher precedence to any other filters that are attached to the same interfaces.

Filter with interface-set match cannot be used on dynamic interface—dynamic interface-set match is not supported. The shared-name of an interface-shared filter can now be populated from the $junos-svlan-interface-set-name variable. This means interface-shared filter can also be attached to dynamic SVLAN interface-set, before which the shared-name could only be taken from the $junos-interface-set-name variable.

To configure an interface-shared filter using a dynamic profile that is used to implement agent-circuit-identifier VLAN household filtering:

1. Access the dynamic profile you want to use.

```
[edit]
user@host# edit dynamic-profiles client-profile
```

2. Specify the interfaces and the unit.

```
[edit dynamic-profiles client-profile]
user@host# edit interfaces demux0 unit $junos-interface-unit
```

3. Specify the family.

```
[edit dynamic-profiles client-profile interfaces demux0 unit “$junos-interface-unit”]
user@host# edit family inet
```

4. Specify the input filter and the filter terms for the interface unit.

```
[edit dynamic-profiles client-profile interfaces demux0 unit “$junos-interface-unit” family inet]
user@host# edit input $junos-input-filter shared-name $junos-interface-set-name precedence
precedence-number
```

5. Specify the output filter and the filter terms for the interface unit.

```
[edit dynamic-profiles client-profile interfaces demux0 unit “$junos-interface-unit” family inet]
user@host# edit input $junos-output-filter shared-name $junos-interface-set-name precedence
precedence-number
```

6. Specify that you want to configure a firewall, and specify the family.

```
[edit dynamic-profiles client-profile]
user@host# edit firewall family inet
```

7. Specify the filter.

```
[edit dynamic-profiles client-profile firewall family inet
user@host# edit filter uid
```

8. Specify that the filter is an interface-shared filter.

```
[edit dynamic-profiles client-profile firewall family inet filter uid]
user@host# set interface-shared
```

```
[edit]
    dynamic-profile {
    client-profile {
        interfaces {
            demux0 {
                unit $junos-interface-unit {
                    family inet {
                        filter {
                            input $junos-input-filter shared-name $junos-interface-set-name
precedence 10;
                        }
                    }
                }
            }
        }
    }
    }

    firewall {
        family inet {
            filter FILTER1 {
                interface-shared;

                term… # the filter’s terms
            }
```

```
        }
    }
```

RELATED DOCUMENTATION

Dynamically Attaching Statically Created Filters for a Specific Interface Family Type | 255

Dynamically Attaching Filters Using RADIUS Variables | 307

Firewall Filters Overview

Example: Dynamic-Profile Parsing

The following example shows the basic dynamic-profile parsing steps for parameterized filters.

1. Read **dynamic-profiles my-svc-prof interface ge-1/0/0 unit 7 family inet filter input** and get the value **my-filt-1gw_UID1022**. The **my-in-filter** variable received the name of the UID (**my-filt-1gw**) from the first service parameter. The name **my-filt-1gw_UID1022** comes from the value of the **my-filt-1gw UID**.
2. Determine whether a static filter called **my-filt-1gw_UID1022** exists. If so, this is the existing classic filter case and not a parameterized filter.
3. Try to read **dynamic-profile my-svc-prof firewall family inet fast-update-filter my-filt-1gw_UID1022**'. If this exists, this is a fast update filter, not a parameterized filter.
4. Try to read **dynamic-profile my-svc-prof firewall family inet filter my-filt-1gw_UID1022**. If this does not exist, return a "filter not found" error.
5. Search for a template named **my-filt-1gw_UID1022**. If it does not exist:
   a. Read the parameterized filter configuration. This adds the match destination address **198.51.100.239** and the policer **my-policer_UID1005** as the action.
   b. Determine whether **my-policer_UID1005** exists. If it does not, read the **dynamic-profile my-svc-prof firewall policer my-policer_UID1005** configuration and create the **my-policer_UID1005** policer.
   c. Compile the **my-filt-1gw_UID1022** filter.
   d. Install **my-filt-1gw_UID1022** as a filter template.
6. Create and install an interface-specific filter reference named my-filt-1gw_UID1022-ge-1/0/0.7-in with **my-filt-1gw_UID1022** as its template.
7. Attach **my-filt-1gw_UID1022-ge-1/0/0.7-in** to interface **ge-1/0/0.7**.

When subsequent sessions are created with the same parameters, the system returns the same **my-filt-1gw_UID1022** filter name. In this case, Step 5 finds the existing filter template and proceeds directly to Step 6.

RELATED DOCUMENTATION

Sample Dynamic-Profile Configuration for Parameterized Filters | 263

Dynamic Profile After UID Substitutions for Parameterized Filters | 266

Example: Firewall Dynamic Profile

In this example, dynamic firewall is configured for subscriber access using Junos IPv4 predefined variables.

The predefined variables equate to RADIUS settings as follows:

Junos OS Predefined Variable	RADIUS VSA Name	RADIUS Attribute Number
$junos-input-filter	Ingress-Policy-Name	26–10
$junos-output-filter	Egress-Policy-Name	26–11

```
dynamic-profiles {
    DynamicFilterProfile {
        interfaces {
            “$junos-interface-ifd-name” {
                unit “$junos-underlying-interface-unit” {
                    family inet {
                        filter {
                            input “$junos-input-filter”;
                            output “$junos-output-filter”;
                        }
                    }
                }
            }
        }
```

```
    }
}
```

NOTE: You must also configure any global firewall parameters.

RELATED DOCUMENTATION

Understanding Dynamic Firewall Filters | 236

Example: Configuring a Filter to Exclude DHCPv6 and ICMPv6 Control Traffic for LAC Subscriber

IN THIS SECTION

- Requirements | 314
- Overview | 314
- Configuration | 315

This example shows how to configure a standard stateless firewall filter that excludes DHCPv6 and ICMPv6 control packets from being considered for idle-timeout detection for tunneled subscribers at the LAC.

Requirements

No special configuration beyond device initialization is required before configuring this example.

Overview

Subscriber access on a LAC can be limited by configuring an idle timeout period that specifies the maximum period of time a subscriber can remain idle after the subscriber session is established. The LAC monitors the subscriber's upstream and downstream data traffic to determine whether the subscriber is inactive. Based on the session accounting statistics. the subscriber is not considered idle as long as data traffic is detected in either direction. When no traffic is detected for the duration of the idle

time out, the subscriber is logged out gracefully similarly to a RADIUS-initiated disconnect or a CLI-initiated logout.

However, after a tunnel is established for L2TP subscribers, all packets through the tunnel at the LAC are treated as data packets. Consequently, the accounting statistics for the session are inaccurate and the subscriber is not considered to be idle as long as DHCPv6 and ICMPv6 control packets are being sent.

You can define a firewall filter for the `inet6` family with terms to match on these control packets. Include the use the `exclude-accounting` terminating action in the filter terms to drop these control packets.

Configuration

IN THIS SECTION

- CLI Quick Configuration | 315
- Configure the Filter | 316
- Results | 318

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 access profile v6-exclude-idle session-options client-idle-timeout 10
set access profile v6-exclude-idle session-options client-idle-timeout-ingress-only

edit firewall family inet6 filter EXCLUDE-ACCT-INET6-FILTER
set interface-specific
set term EXCLUDE-ACCT-DHCP-INET6 from next-header udp
set term EXCLUDE-ACCT-DHCP-INET6 from source-port 546
set term EXCLUDE-ACCT-DHCP-INET6 from source-port 547
set term EXCLUDE-ACCT-DHCP-INET6 from destination-port 546
set term EXCLUDE-ACCT-DHCP-INET6 from destination-port 547
set term EXCLUDE-ACCT-DHCP-INET6 then count exclude-acct-dhcpv6
set term EXCLUDE-ACCT-DHCP-INET6 then exclude-accounting

set term EXCLUDE-ACCT-ICMP6 from next-header icmp6
```

```
set term EXCLUDE-ACCT-ICMP6 from icmp-type router-solicit
set term EXCLUDE-ACCT-ICMP6 from icmp-type neighbor-solicit
set term EXCLUDE-ACCT-ICMP6 from icmp-type neighbor-advertisement
set term EXCLUDE-ACCT-ICMP6 then count exclude-acct-icmpv6
set term EXCLUDE-ACCT-ICMP6 then exclude-accounting

set term default then accept

top
edit dynamic-profiles pppoe-dynamic-profile interfaces pp0 unit "$junos-interface-unit"
set family inet6 filter input EXCLUDE-ACCT-INET6-FILTER
set family inet6 filter output EXCLUDE-ACCT-INET6-FILTER
set actual-transit-statistics
```

Configure the Filter

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in theCLI User Guide.

To configure the filter:

1. Set the idle timeout for subscriber sessions..

```
[edit access profile v6-exclude-idle]
user@host# set session-options client-idle-timeout 10
```

2. Specify the idle timeout applies only to ingress traffic.

```
[edit access profile v6-exclude-idle]
user@host# set session-options client-idle-timeout-ingress-only
```

3. Define the firewall filter term that excludes the DHCPv6 control packets from accounting statistics.

   a. Specify a match on packets with the first Next Header field set to UDP (17).

```
[edit firewall family inet6 filter EXCLUDE-ACCT-INET6-FILTER]
user@host# set term EXCLUDE-ACCT-DHCP-INET6 from next-header udp
```

b. Specify a match on packets with a source port of 546 or 547 (DHCPv6).

```
[edit firewall family inet6 filter EXCLUDE-ACCT-INET6-FILTER]
user@host# set term EXCLUDE-ACCT-DHCP-INET6 from source-port 546
user@host# set term EXCLUDE-ACCT-DHCP-INET6 from source-port 547
```

c. Specify a match on packets with a DHCP destination port of 546 or 547 (DHCPv6).

```
[edit firewall family inet6 filter EXCLUDE-ACCT-INET6-FILTER]
user@host# set term EXCLUDE-ACCT-DHCP-INET6 from destination-port 546
user@host# set term EXCLUDE-ACCT-DHCP-INET6 from destination-port 547
```

d. Count the matched DHCPv6 packets.

```
[edit firewall family inet6 filter EXCLUDE-ACCT-INET6-FILTER]
user@host# set term EXCLUDE-ACCT-DHCP-INET6 then count exclude-acct-dhcpv6
```

e. Exclude the matched DHCPv6 packets from accounting statistics.

```
[edit firewall family inet6 filter EXCLUDE-ACCT-INET6-FILTER]
user@host# set term EXCLUDE-ACCT-DHCP-INET6 then exclude-accounting
```

4. Define the firewall filter term that excludes the ICMPv6 control packets from accounting statistics.

a. Specify a match on packets with the first Next Header field set to ICMPv6 (58).

```
[edit firewall family inet6 filter EXCLUDE-ACCT-INET6-FILTER]
user@host# set term EXCLUDE-ACCT-ICMP6 from next-header icmp6
```

b. Specify a match on packets with an ICMPv6 message type.

```
[edit firewall family inet6 filter EXCLUDE-ACCT-INET6-FILTER]
user@host# set term EXCLUDE-ACCT-ICMP6 from icmp-type router-solicit
user@host# set term EXCLUDE-ACCT-ICMP6 from icmp-type neighbor-solicit
user@host# set term EXCLUDE-ACCT-ICMP6 from icmp-type neighbor-advertisement
```

c. Count the matched ICMPv6 packets.

```
[edit firewall family inet6 filter EXCLUDE-ACCT-INET6-FILTER]
user@host# set term EXCLUDE-ACCT-ICMP6 then count exclude-acct-icmpv6
```

d. Exclude the matched ICMPv6 packets from accounting statistics.

```
[edit firewall family inet6 filter EXCLUDE-ACCT-INET6-FILTER]
user@host# set term EXCLUDE-ACCT-DHCP-INET6 then exclude-accounting
```

5. Define the default filter term to accept all other packets.

```
[edit firewall family inet6 filter EXCLUDE-ACCT-INET6-FILTER]
user@host# set term default then accept
```

6. Configure the dynamic profile to apply the filter to input and output interfaces for the `inet6` family.

```
[edit dynamic-profiles pppoe-dynamic-profile interfaces pp0 unit "$junos-interface-unit"]
user@host# set family inet6 filter input EXCLUDE-ACCT-INET6-FILTER
user@host# set family inet6 filter output EXCLUDE-ACCT-INET6-FILTER
```

7. Enable subscriber management accurate accounting.

```
[edit dynamic-profiles pppoe-dynamic-profile interfaces pp0 unit "$junos-interface-unit"]
user@host# set actual-transit-statistics
```

Results

From configuration mode, confirm your configuration by entering the `show access`, `show firewall`, and `show dynamic-profiles` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show access
profile v6-exclude-idle {
    session-options {
        client-idle-timeout 10;
        client-idle-timeout-ingress-only;
```

```
    }
}
```

```
user@host# show firewall
family inet6 {
    filter EXCLUDE-ACCT-INET6-FILTER {
        interface-specific;
        term EXCLUDE-ACCT-DHCP-INET6 {
            from {
                next-header udp;
                 source-port [ 546 547 ];
                destination-port [ 546 547 ];
            }
            then {
                count exclude-acct-dhcpv6;
                exclude-accounting
            }
        }
        term EXCLUDE-ACCT-ICMP6 {
            from {
                next-header icmp6;
                 icmp-type [ router-solicit neighbor-solicit neighbor-advertisement ]
            }
            then {
                count exclude-acct-icmpv6;
                exclude-accounting;
            }
        }
        term default {
            then accept;
        }
    }
}
```

```
user@host# show dynamic-profiles
pppoe-dynamic-profile {
    interfaces {
        pp0 {
            unit "$junos-interface-unit" {
                actual-transit-statistics;
```

```
                family inet6 {
                    filter {
                        input EXCLUDE-ACCT-INET6-FILTER;
                        output EXCLUDE-ACCT-INET6-FILTER;
                    }
                }
            }
        }
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

RELATED DOCUMENTATION

- Classic Filters Overview | **239**
- Dynamically Attaching Statically Created Filters for a Specific Interface Family Type | **255**
- Understanding How to Use Standard Firewall Filters
- Firewall Filter Terminating Actions
- Firewall Filter Match Conditions for IPv6 Traffic

CHAPTER 20

Using Ascend Data Filters to Implement Firewalls Based on RADIUS Attributes

IN THIS CHAPTER

- Ascend-Data-Filter Policies for Subscriber Management Overview | 321
- Ascend-Data-Filter Attribute Fields | 324
- Dynamically Applying Ascend-Data-Filter Policies to Subscriber Sessions | 328
- Example: Configuring Dynamic Ascend-Data-Filter Support for Subscriber Access | 331
- Example: Configuring Static Ascend-Data-Filter Support for Subscriber Access | 336
- Verifying and Managing Dynamic Ascend-Data-Filter Policy Configuration | 342

Ascend-Data-Filter Policies for Subscriber Management Overview

IN THIS SECTION

- Filter Naming Conventions | 322
- Use of Multiple Sessions with Ascend-Data-Filters on an Interface | 323
- Optional ADF Filter Requirement for Some Subscribers | 323

Subscriber management enables you to use Ascend-Data-Filters to create policies for subscriber traffic. An Ascend-Data-Filter is a binary value that is configured on the RADIUS server. The filter contains rules that specify match conditions for traffic and an action for the router to perform (such as accept or discard the traffic). The match conditions might include the source and destination IP address or port, the protocol, the filter direction, the traffic class, and policer information.

Subscriber management uses a dynamic profile to obtain the Ascend-Data-Filter attribute (RADIUS attribute 242) from the RADIUS server and apply the policy to a subscriber session. Dynamic profiles support Ascend-Data-Filters for `inet` and `inet6` family types, and both families can be present in a

dynamic profile. You include Junos OS predefined variables in the dynamic profiles — `$junos-adf-rule-v4` for family `inet` and `$junos-adf-rule-v6` for `inet6`. The Ascend-Data-Filter attribute can include rules for both address families. The predefined variables map the Ascend-Data-Filter rules for the respective family to the Junos OS *firewall filter* process. A firewall filter is created and attached to the subscriber's *logical interface*.

You can also configure a static Ascend-Data-Filter by manually entering the required binary data as a hexadecimal string in a dynamic profile. A statically configured Ascend-Data-Filter in a dynamic profile takes precedence over an Ascend-Data-Filter attribute that is received from RADIUS. The static method is time-consuming to configure; it is typically used only for testing purposes.

The Ascend-Data-Filter attribute is supported in RADIUS Access-Accept and Change of Authorization (CoA) messages.

CoA updates existing filters based on the Ascend-Data-Filter Type field, as shown in the following list:

- If the Type field is 1, IPv4 rules are updated and IPv6 rules are unchanged. The opposite is true if the Type field is 3.
- If both Type 1 and 3 are specified, then all rules are updated.
- If the CoA has no Ascend-Data-Filter rules, then the existing rules are unchanged.

NOTE: On Junos OS Evolved:

- ADF rule do not support the counter action.
- IPv6 ADF rules support the entire prefix length range (128 bits) for both source and destination prefixes.
- You can configure up to 100 subscriber ADF rules (10 `inet` and 10 `inet6` per subscriber), which means a total of 2000 access control list (ACL) entries.
- When you apply ADF rules to subscribers, the last rule must have the drop (discard) action.
- ADF rules are not supported in egress direction.

Filter Naming Conventions

Each Ascend-Data-Filter has a unique name, which is assigned by the dynamic firewall process, dfwd. The assigned names are displayed in the results of the **show subscriber extensive** and **show firewall** commands. Ascend-Data-Filters use the following naming convention:

`__junos_adf_`*`session#-interfacename-family-direction`*

For example:

```
__junos_adf_33847-ge/1/0/4.53-init-in
```

Each Ascend-Data-Filter rule maps to a single term, and the term names are simply `t0`, `t1`, ..., `tn`. If you configure the **counter** option, the router adds a count action to each term that is created. The counter names are a combination of the the term names with `-cnt` appended. For example `t0-cnt` and `t1-cnt`.

Use of Multiple Sessions with Ascend-Data-Filters on an Interface

An interface can have multiple subscriber sessions, each session using its own Ascend-Data-Filter rules. When an Ascend-Data-Filter is applied to a subscriber session, the rules are created independently of any other filters and are added to the interface filter list. The Ascend-Data-Filter rules for the other sessions on the same interface are also added to the filter list. All packets that are processed for the interface must go through all filters, and the filters are applied according to the precedence you set.

Because the filter list can be a combination of several rules, you must consider how the multiple filters coexist. You must ensure that the filters are designed and applied correctly in order to provide the desired filtering and resulting action. For example, a session might have a filter that accepts traffic from Subscriber-A and discards all other traffic. However, a second session on the same interface might have a filter that accepts traffic from Subscriber-B only and discards other traffic. When the two filters are combined in the filter list, traffic from Subscriber-B is discarded by the first filter, and traffic from Subscriber-A is discarded by the second filter. As a result, no traffic is accepted on the interface because the two filters essentially cancel out each other and discard all traffic.

Optional ADF Filter Requirement for Some Subscribers

When you include either of the predefined variables—`$junos-adf-rule-v4` or `$junos-adf-rule-v6`—in the dynamic profile, by default the RADIUS reply message must include the Ascend-Data-Filter attribute (RADIUS attribute 242) for each subscriber. If the attribute is not included, the router reports an error.

A service provider might apply the same dynamic profile to a mixed pool of subscribers, such that the attribute is included by RADIUS for some of the subscribers and is not included for others. By default, the router returns an error for each of the subscribers without the attribute, consuming system resources. You can configure the dynamic profile to accommodate such a mixture of subscribers by making the attribute requirement optional. To do so, and to suppress attribute error reporting, specify the not-mandatory option with the `adf` statement at the `[edit dynamic-profiles` *`profile-name`* `interfaces` *`interface-name`* `unit` *`logical-unit-number`* `family` *`family`* `filter]` hierarchy level. With this configuration, the Ascend-Data-filter is simply not created when the Ascend-Data-Filter attribute is not present.

RELATED DOCUMENTATION

Dynamically Applying Ascend-Data-Filter Policies to Subscriber Sessions | 328

Ascend-Data-Filter Attribute Fields

Table 30 on page 324 provides information about the fields used in the Ascend-Data-Filter attribute (RADIUS attribute 242) and how the fields map to Junos OS filter functions. The table lists the fields in the order in which they occur in the Ascend-Data-Filter attribute.

Table 30: Ascend-Data-Filter Attribute Fields

Action or Classifier	Format	Value	Junos OS Filter Function
Type	1 byte	• 1 = IPv4 • 3 = IPv6	
Filter or forward	1 byte	• 0 = filter • 1 = forward	• 0 = maps to `discard` action • 1 = maps to `accept` action
Indirection	1 byte	• 0 = egress • 1 = ingress	• 0 = adds egress terms to the output filter • 1= adds ingress terms to the input filter
Spare	1 byte	–	–
Source IP address	IPv4 = 4 bytes IPv6 = 16 bytes	IP address of the source interface	• 0 = no mapping performed • From `source-address` *address* entry added to term
Destination IP address	IPv4 = 4 bytes IPv6 = 16 bytes	IP address of the destination interface	• 0 = no mapping performed • From `destination-address` *address* entry added to term

Table 30: Ascend-Data-Filter Attribute Fields *(Continued)*

Action or Classifier	Format	Value	Junos OS Filter Function
Source IP prefix	1 byte	• Type 1 = Number of leading zeros in the wildcard mask • Type 3 = Higher order contiguous bits of the address that make up the network portion of the address	• 0 = no mapping performed • From `source-address` *`prefix`* entry added to term
Destination IP prefix	1 byte	• Type 1 = Number of leading zeros in the wildcard mask • Type 3 = Higher order contiguous bits of the address that make up the network portion of the address	• 0 = no mapping performed • From `destination-address` *`prefix`* entry added to term
Protocol	1 byte	Protocol type	• 0 = no mapping performed • IPv4 = from `protocol` *`number`* added to term • IPv6 = from `next-header` *`number`* added to term
Established	1 byte	Not implemented	Not implemented
Source port	2 bytes	Port number of the source port	From `source-port` *`x`* `-` *`y`* entry added to term
Destination port	2 bytes	Port number of the destination port	From `destination-port` *`x`* `-` *`y`* entry added to term

Table 30: Ascend-Data-Filter Attribute Fields *(Continued)*

Action or Classifier	Format	Value	Junos OS Filter Function
Source port qualifier	1 byte	• 0 = no compare • 1 = less than • 2 = equal to • 3 = greater than • 4 = not equal to	• 0 = no mapping performed • 1 – 3 = mapped to corresponding option • • 4 = mapped to `except` match option
Destination port qualifier	1 byte	• 0 = no compare • 1 = less than • 2 = equal to • 3 = greater than • 4 = not equal to	• 0 = no mapping performed • 1 – 3 = mapped to corresponding match option • • 4 = mapped to `except` match option
Reserved	2 bytes	Not used	Not used
Marking value	1 byte	• For IPv4 = Type of Service (ToS) • For IPv6 = Differentiated Services Code Point (DSCP)	Not implemented
Marking mask	1 byte	0 = no packet marking	Not implemented

Table 30: Ascend-Data-Filter Attribute Fields *(Continued)*

Action or Classifier	Format	Value	Junos OS Filter Function
Traffic class	1–41 bytes	• 0 = no traffic class (required if there is no profile) • First byte specifies the length of the ASCII name of the traffic class • Traffic class must be statically configured • Name can optionally be null terminated, which consumes 1 byte • If a name is given, it must match one of the default forwarding classes (such as best-effort) or the name of a forwarding class configured under the `[edit class-of-service scheduler-maps` *`map-name`*`]` stanza.	Maps to the forwarding class name. The action `forwarding-class` *`name`* is added to term.

Table 30: Ascend-Data-Filter Attribute Fields *(Continued)*

Action or Classifier	Format	Value	Junos OS Filter Function
Rate-limit profile	1–41 bytes	• 0 = no rate limit (required if there is no profile) • First byte specifies the length of the ASCII, followed by the ASCII name of the profile • Profile must be statically configured • Name can optionally be null terminated, which consumes 1 byte • If a name is given, it must match the name of one of the firewall policers that is configured under the `[edit firewall]` stanza.	Maps to the `policer` *`policer-name`* action modifier of the same name. The action `policer` *`name`* is added to term.

RELATED DOCUMENTATION

Ascend-Data-Filter Policies for Subscriber Management Overview | 321

Dynamically Applying Ascend-Data-Filter Policies to Subscriber Sessions

Subscriber management enables you to use dynamic profiles to dynamically apply policies that are defined in Ascend-Data-Filters (RADIUS attribute 242) to subscriber sessions. The dynamic profiles include a Junos OS predefined variable that maps the rules and actions defined in the Ascend-Data-Filter to Junos OS features. The RADIUS administrator configures the Ascend-Data-Filter on the RADIUS server in a separate operation.

Subscriber management dynamic profiles use the following Junos OS predefined variables to map family-specific Ascend-Data-Filter rules to Junos OS filter functionality:

- `$junos-adf-rule-v4`—Used for IPv4 family `inet`.

- `$junos-adf-rule-v6`—Used for IPv6 family `inet6`.

To configure a dynamic profile to dynamically apply the policy defined by an Ascend-Data-Filter to a subscriber session:

1. Specify the dynamic profile in which you want to include the Ascend-Data-Filter. Specify the interface, the logical unit number, and the family type.

```
[edit]
user@host# edit dynamic-profiles profile-name interfaces interface-name unit logical-unit-number family family
```

2. Specify that you want to include an Ascend-Data-Filter in the dynamic profile.

```
[edit dynamic-profiles profile-name interfaces interface-name unit logical-unit-number family family]
user@host# edit filter adf
```

3. Specify the Junos OS predefined variable that maps the Ascend-Data-Filter actions to Junos OS filter functionality. Use the variable that corresponds to the specified family type.

```
[edit dynamic-profiles profile-name interfaces interface-name unit logical-unit-number family family filter adf]
user@host# set rule ($junos-adf-rule-v4 | $junos-adf-rule-v6)
```

NOTE: You can also statically configure the Ascend-Data-Filter in this step by entering the filter in hexadecimal format, rather than use a predefined variable. You might use a static filter for testing purposes.

4. (Optional) Suppress error-reporting in the event the RADIUS reply messages do not include the Ascend-Data-Filter attribute.

```
[edit dynamic-profiles profile-name interfaces interface-name unit logical-unit-number family family filter adf]
user@host# set not-mandatory
```

5. (Optional) Enable the counter feature. The counter increments each time a packet matches the rule.

```
[edit dynamic-profiles profile-name interfaces interface-name unit logical-unit-number family
family filter adf]
user@host# set counter
```

6. (Optional) Specify the input precedence used to establish the order in which filters on the interface are applied. A lower precedence value equals a higher precedence. The precedence relates to other dynamic filters configured on the same interface.

```
[edit dynamic-profiles profile-name interfaces interface-name unit logical-unit-number family
family filter adf]
user@host# set input-precedence precedence
```

7. (Optional) Specify the output precedence used to establish the order in which filters on the interface are applied. A lower precedence value equals a higher precedence. The precedence relates to other dynamic filters configured on the same interface.

```
[edit dynamic-profiles profile-name interfaces interface-name unit logical-unit-number family
family filter adf]
user@host# set output-precedence precedence
```

RELATED DOCUMENTATION

- Ascend-Data-Filter Policies for Subscriber Management Overview | 321
- Ascend-Data-Filter Attribute Fields | 324
- Verifying and Managing Dynamic Ascend-Data-Filter Policy Configuration | 342
- Example: Configuring Dynamic Ascend-Data-Filter Support for Subscriber Access | 331
- Example: Configuring Static Ascend-Data-Filter Support for Subscriber Access | 336

Example: Configuring Dynamic Ascend-Data-Filter Support for Subscriber Access

IN THIS SECTION

- Requirements | 331
- Overview | 331
- Configuration | 332
- Verification | 334

This example shows how to configure support for dynamic Ascend-Data-Filter policies.

Requirements

- Ensure that the Ascend-Data-Filter has been configured on the RADIUS server.
- Create the dynamic profile. See *Dynamic Profiles Overview*.
- Configure RADIUS support. See *RADIUS Servers and Parameters for Subscriber Access*.

Overview

Ascend-Data-Filters are configured on a RADIUS server, and contain rules that create policies. Subscriber management uses a dynamic profile to obtain the Ascend-Data-Filter attribute (RADIUS attribute 242) from the RADIUS server and apply the policy to a subscriber session.

- Specify the dynamic profile to use to apply the Ascend-Data-Filter policy to the subscriber session.
- Specify the Junos OS predefined variable that maps the Ascend-Data-Filter rules to Junos OS filter functionality.
- Configure optional settings, which include counting the rule usage and setting the precedence order for the filter.

Configuration

IN THIS SECTION

- Procedure | 332

Procedure

Step-by-Step Procedure

To configure dynamic Ascend-Data-Filter support:

1. Specify the dynamic profile in which you want to include the Ascend-Data-Filter, and configure the interface, the logical unit number, and the family type.

```
[edit]
user@host# edit dynamic-profiles adf-profile-v4 interfaces $junos-interface-ifd-name unit
$junos-underlying-interface-unit family inet
```

2. Specify that you want to include an Ascend-Data-Filter in the dynamic profile and provide the Junos OS predefined variable as the rule that maps the Ascend-Data-Filter actions to Junos OS filter functionality.

```
[edit dynamic-profiles adf-profile-v4 interfaces “$junos-interface-ifd-name” unit “$junos-
underlying-interface-unit” family inet]
user@host# set filter adf rule $junos-adf-rule-v4
```

3. Enable the counter for the rule.

```
[edit dynamic-profiles adf-profile-v4 interfaces “$junos-interface-ifd-name” unit “$junos-
underlying-interface-unit” family inet]
user@host# set filter adf counter
```

4. Specify the precedence for received packets on the interface.

```
[edit dynamic-profiles adf-profile-v4 interfaces “$junos-interface-ifd-name” unit “$junos-
underlying-interface-unit” family inet]
user@host# set filter adf input-precedence 75
```

5. Specify the precedence for transmitted packets on the interface.

```
[edit dynamic-profiles adf-profile-v4 interfaces “$junos-interface-ifd-name” unit “$junos-
underlying-interface-unit” family inet]
user@host# set filter adf output precedence 80
```

Results

From configuration mode, confirm your configuration by entering the `show dynamic-profiles` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show dynamic-profiles
...
adf-profile-v4 {
    interfaces {
        "$junos-interface-ifd-name" {
            unit "$junos-underlying-interface-unit" {
                family inet {
                    filter {
                        adf {
                            rule "$junos-adf-rule-v4";
                            counter;
                            input-precedence 75;
                            output-precedence 80;
                        ...
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Verifying that Dynamic Ascend-Data-Filter Rules Are Applied to Subscriber Sessions | **334**
- Verifying Dynamic Ascend-Data-Filter Usage | **335**

To confirm that the configuration is working properly, perform these tasks:

Verifying that Dynamic Ascend-Data-Filter Rules Are Applied to Subscriber Sessions

Purpose

Verify that the Ascend-Data-Filter rules were attached to the subscriber.

Action

From operational mode, enter the `show subscribers extensive` command.

```
user@host>show subscribers extensive
Type: DHCP
User Name: user1-adf
IP Address: 192.168.1.10
IP Netmask: 255.255.255.0
Logical System: default
Routing Instance: default
Interface: ge-1/0/0.0
Interface type: Static
Dynamic Profile Name: adf-profile-v4
MAC Address: 00:00:5E:00:53:01
State: Active
Radius Accounting ID: 5
Login Time: 2010-08-12 14:06:27 PDT
ADF IPv4 Input Filter Name: __junos_adf_5-ge-1/0/0.0-inet-in
                    Rule 0: 0101010000000000d87f9200001800000000000000000000
                            from {
                                destination-address 198.51.100.146.0/24;
                            }
```

```
                then {
                    accept;
                }
            Rule 1: 0100010000000000000000000000060000000001900020000
                from {
                    protocol 6;
                    destination-port 25;
                }
                then {
                    discard;
                }
            Rule 2: 0101010000000000000000000000000000000000000000000
                then {
                    accept;
                }
```

Meaning

The output shows the information for the dynamic profile, including Ascend-Data-Filter rules. Verify the following information:

- The User Name field indicates the correct subscriber.
- The Dynamic Profile Name field is correct for the subscriber.
- The correct Ascend-Data-Filter rules are applied to the subscriber. The display shows the rules that are configured on the RADIUS server.

Verifying Dynamic Ascend-Data-Filter Usage

Purpose

Verify usage of the dynamic Ascend-Data-Filter. Counter statistics are displayed when the `counter` option is configured for the `adf` command in the dynamic profile.

Action

From operational mode, enter the `show firewall` command.

```
user@host> show firewall

Filter: __junos_adf_5-ge-1/0/0.0-inet-in
```

```
Counters:
Name                        Bytes             Packets
t0-cnt                      32758                  22
t1-cnt                      22199                  15
t2-cnt                      21723                  14
```

Meaning

The output shows the name of the filter and lists the counter activity. If the `counter` option is not configured, the output displays only the filter name.

RELATED DOCUMENTATION

Ascend-Data-Filter Policies for Subscriber Management Overview | 321

Dynamically Applying Ascend-Data-Filter Policies to Subscriber Sessions | 328

Example: Configuring Static Ascend-Data-Filter Support for Subscriber Access

IN THIS SECTION

- Requirements | 337
- Overview | 337
- Configuration | 337
- Verification | 340

This example shows how to configure support for static Ascend-Data-Filter policies. In a static configuration, you manually configure the Ascend-Data-Filter as part of the dynamic profile configuration. This procedure differs from dynamic configuration, in which the Ascend-Data-Filter is defined on the RADIUS server and then subscriber management uses a predefined variable to map the

Ascend-Data-Filter rules to Junos OS filter functionality. Because creating a static Ascend-Data-Filter configuration can be labor-intensive, you might typically use this method for testing purposes.

Requirements

- Create the dynamic profile. See *Dynamic Profiles Overview*.
- Configure RADIUS support. See *RADIUS Servers and Parameters for Subscriber Access*.

Overview

Ascend-Data-Filters contain rules that create policies. Subscriber management uses a dynamic profile to apply the policy to a subscriber session. You manually configure the Ascend-Data-Filter as part of the dynamic policy.

- Specify the dynamic profile to use to apply the Ascend-Data-Filter policy to the subscriber session.
- Configure the Ascend-Data-Filter.
- Configure optional settings, which include counting the rule usage and setting the precedence for received and transmitted traffic.

Configuration

IN THIS SECTION

Procedure | 337
Results | 339

Procedure

Step-by-Step Procedure

To configure static Ascend-Data-Filter support:

1. Specify the dynamic profile in which you want to create the Ascend-Data-Filter, and configure the interface, the logical unit number, and the family type.

```
[edit]
user@host# edit dynamic-profiles adf-profile-v4 interfaces $junos-interface-ifd-name unit
$junos-underlying-interface-unit family inet
```

2. Configure the Ascend-Data-Filter. Enclose the filter values within quotation marks. You can configure multiple Ascend-Data-Filter rules in the same dynamic profile.

```
[edit dynamic-profiles adf-profile-v4 interfaces “$junos-interface-ifd-name” unit “$junos-
underlying-interface-unit” family inet]
user@host# set filter adf rule “01000100 CB007100 00000000 18000000 00000000 00000000”
```

3. Enable the counter for the rule.

```
[edit dynamic-profiles adf-profile-v4 interfaces “$junos-interface-ifd-name” unit “$junos-
underlying-interface-unit” family inet]
user@host# set filter adf counter
```

4. Specify the precedence for received packets on the interface.

```
[edit dynamic-profiles adf-profile-v4 interfaces “$junos-interface-ifd-name” unit “$junos-
underlying-interface-unit” family inet]
user@host# set filter adf input-precedence 80
```

5. Specify the precedence for transmitted packets on the interface.

```
[edit dynamic-profiles adf-profile-v4 interfaces “$junos-interface-ifd-name” unit “$junos-
underlying-interface-unit” family inet]
user@host# set filter adf output precedence 85
```

Results

From configuration mode, confirm your configuration by entering the `show dynamic-profiles` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show dynamic-profiles
...
adf-profile-v4 {
    interfaces {
        "$junos-interface-ifd-name" {
            unit "$junos-underlying-interface-unit" {
```

```
                family inet {
                    filter {
                        adf {
                            rule "01000100 CB007100 00000000 18000000 00000000 00000000";
                            counter;
                            input-precedence 80;
                            output-precedence 85;
                        ...
```

If you are done configuring the device, enter `commit` from configuration mode.

Results

The Ascend-Data-Filter rule defined in Step "2" on page 338 of the procedure configures an input policy that filters all packets from network 203.0.113.0 with wildcard mask 255.255.255.0 to any destination.

Table 31 on page 339 lists the values specified in the Ascend-Data-Filter rule.

Table 31: Ascend-Data-Filter Rule

Action or Classifier	Hex Value	Junos OS Filter Function
Type	01	IPv4
Forward	00	Forward
Indirection	01	Ingress
Spare	00	None
Source IP address	CB007100	203.0.113.0
Destination IP address	00000000	Any
Source IP mask	18	24 (255.255.255.0)
Destination IP mask	00	0 (0.0.0.0)

Table 31: Ascend-Data-Filter Rule *(Continued)*

Action or Classifier	Hex Value	Junos OS Filter Function
Protocol	00	None
Established	00	None
Source port	0000	None
Destination port	0000	None
Source port qualifier	00	None
Destination port qualifier	00	None
Reserved	0000	None

Verification

IN THIS SECTION

- Verifying that Static Ascend-Data-Filter Rules are Applied to Subscriber Sessions | **340**
- Verifying Static Ascend-Data-Filter Usage | **342**

To confirm that the configuration is working properly, perform these tasks:

Verifying that Static Ascend-Data-Filter Rules are Applied to Subscriber Sessions

Purpose

Verify that the Ascend-Data-Filter rules you manually configured were attached to the subscriber.

Action

From operational mode, enter the `show subscribers extensive` command.

```
user@host>show subscriber extensive
Type: DHCP
User Name: user1-adf
IP Address: 192.168.1.10
IP Netmask: 255.255.255.0
Logical System: default
Routing Instance: default
Interface: ge-1/0/0.0
Interface type: Static
Dynamic Profile Name: adf-profile-v4
MAC Address: 00:00:5E:00:53:01
State: Active
Radius Accounting ID: 5
Login Time: 2010-08-12 14:06:27 PDT
ADF IPv4 Input Filter Name: __junos_adf_5-ge-1/0/0.0-inet-in
                    Rule 0: 01000100CB007100000000001800000000000000000000
                            from {
                                destination-address 203.0.113.0/24;
                            }
                            then {
                                accept;
                            }
```

Meaning

The output shows the information for the dynamic profile, including Ascend-Data-Filter rules. Verify the following information:

- The User Name field indicates the correct subscriber.
- The Dynamic Profile Name field is correct for the subscriber.
- The correct static Ascend-Data-Filter rule is applied to the subscriber.

Verifying Static Ascend-Data-Filter Usage

Purpose

Verify usage of the static Ascend-Data-Filter. Counter statistics are displayed when the `counter` option is configured for the `adf` command in the dynamic profile.

Action

From operational mode, enter the `show firewall` command.

```
user@host> show firewall

Filter: __junos_adf_5-ge-1/0/0.0-inet-in
Counters:
Name                          Bytes              Packets
t0-cnt                        32758                   22
```

Meaning

The output shows the name of the filter and the lists counter activity. If the `counter` option is not configured, the output displays only the filter name.

RELATED DOCUMENTATION

Ascend-Data-Filter Policies for Subscriber Management Overview | 321

Dynamically Applying Ascend-Data-Filter Policies to Subscriber Sessions | 328

Verifying and Managing Dynamic Ascend-Data-Filter Policy Configuration

IN THIS SECTION

Purpose | 343

- Action | 343

Purpose

View or manage information for Ascend-Data-Filters.

Action

- To display statistics for Ascend-Data-Filters:

```
user@host> show firewall
```

- To display firewall log information:

```
user@host> show subscribers extensive
```

- To clear filter counters:

```
user@host> clear firewall all
```

RELATED DOCUMENTATION

Ascend-Data-Filter Policies for Subscriber Management Overview | 321

Dynamically Applying Ascend-Data-Filter Policies to Subscriber Sessions | 328

CHAPTER 21

Configuring Fast Update Filters to Provide More Efficient Processing Over Classic Static Filters

IN THIS CHAPTER

- Fast Update Filters Overview | 344
- Basic Fast Update Filter Syntax | 348
- Configuring Fast Update Filters | 349
- Example: Configuring Fast Update Filters for Subscriber Access | 351
- Match Conditions and Actions in Fast Update Filters | 352
- Configuring the Match Order for Fast Update Filters | 354
- Fast Update Filter Match Conditions | 355
- Fast Update Filter Actions and Action Modifiers | 356
- Configuring Terms for Fast Update Filters | 357
- Configuring Filters to Permit Expected Traffic | 359
- Avoiding Conflicts When Terms Match | 360
- Associating Fast Update Filters with Interfaces in a Dynamic Profile | 367

Fast Update Filters Overview

IN THIS SECTION

- Fast Update Filter Components | 345
- Fast Update Filter Processing | 346
- Fast Update Filter Names | 346
- Guidelines for Creating and Applying Fast Update Filters | 347

Fast update filters provide more efficient filter processing over classic static filters when dynamic services are implemented for multiple subscribers that share the same logical interface.

Fast update filters support subscriber-specific filter values, as opposed to classic filters, which are interface-specific. Fast update filters allow individual filter terms, or rules, to be added or removed from filters without requiring the router to recompile the filter after each modification—terms are added and removed when subscriber services are added and removed.

Using the fast update filters feature involves three distinct operations:

1. Creating the filter—You define fast update filters under the `[edit dynamic-profiles` *`profile-name`* `firewall family` *`family`*`]` hierarchy. The `dynamic-profiles` stanza enables you to use dynamic variables to create subscriber-specific configurations for the filter's match terms. See "Configuring Fast Update Filters" on page 349.

2. Associating the filter with a dynamic profile—You use the `[edit dynamic-profiles` *`profile-name`* `interface` *`interface-name`* `unit` *`unit-number`* `family` *`family`* hierarchy to associate the filter with a dynamic profile. This is the same procedure used for classic filters. See "Associating Fast Update Filters with Interfaces in a Dynamic Profile" on page 367.

3. Attaching the filter to an interface—When a subscriber logs in, the dynamic profile instantiates the subscriber session and applies the properties of the profile, including the fast update filter, to the session interface. This is the same procedure used for classic filters. Also, similar to classic filters, the name of fast update filters can be provided in a user's RADIUS file.

When a dynamic profile instantiates a subscriber session and applies a fast update filter, the router verifies that the filter is not already present on the session interface. If the filter is not present, the router adds the filter. If the filter is already present on the interface, the router simply adds any new terms that are not in the existing filter. This procedure is reversed when subscriber sessions are deleted. Any terms that were added by a session are then removed when the session is deleted. The filter is deleted when the last subscriber session is deleted.

NOTE: You can optionally specify that a term can be added only once and cannot be modified. See "Match Conditions and Actions in Fast Update Filters" on page 352.

This overview covers:

Fast Update Filter Components

When creating a fast update filter, you define one or more terms that specify the filtering criteria and the action to take when a match occurs.

Each term consists of the following components:

- Match condition—Specifies values or fields that the packet must contain. You can match a maximum of five fields in a fast update filter. A match condition can contain a single value or range. This differs from classic filters, in which terms can have multiple values. However, you can use additional terms to specify multiple ranges. "Fast Update Filter Match Conditions" on page 355 lists the supported match conditions for fast update filters. The order in which the terms appear in a fast update filter is not important, because the router examines the most specific term first. (Classic filters examine the terms in the order in which the terms are listed.)

- Action—Specifies what to do when a packet matches the match condition. If no action is specified for a term, the default action is to accept the packet. "Fast Update Filter Actions and Action Modifiers" on page 356 lists the supported actions for fast update filters.

Terms that are added to the filter during session instantiation must have a unique set of match conditions. Two terms overlap, or conflict, if a packet can match both sets of conditions—as a result, there are two different actions for the packet. You can ensure that terms are unique by using the `$junos-subscriber-ip-address` variable as the `source-address` (for an input filter) or `destination-address` (for an output filter) in the `from` statement. You must then supply the `source-address` or `destination-address` condition, as appropriate, as the first condition in the `match-order` statement.

Fast Update Filter Processing

You must use the `match-order` statement to explicitly specify the order in which the router examines filter match conditions. Also, the router examines only those conditions that you include in the `match-order` statement. When a fast update filter contains multiple terms, the router compares a packet against the terms starting with the most specific condition first. When the packet first matches a condition, the router performs the action defined in the term to either accept or reject the packet, and then no other terms are evaluated. If the router does not find a match between the packet and first term, it then compares the packet to the next term in the filter. The router continues to compare the packet to the next specified term until a match is found. If there is no match after all terms have been examined, the router silently drops the packet.

You can specify a precedence (from 0 through 255) for input and output filters within a dynamic profile to force filter processing in a particular order. Setting a lower precedence value for a filter gives it a higher precedence within the dynamic profile. Filters with lower precedence values are applied to interfaces before filters with higher precedence values. A precedence of zero (the default) gives the filter the highest precedence. If no precedence is specified, the filter receives a precedence of zero (highest precedence). Filters with matching precedence (zero or otherwise) are applied in random order.

Fast Update Filter Names

When a filter is attached to an interface, the router first searches for a classic filter with the specified name, and then uses the classic filter. If no classic filter exists with that name, the router then searches in the dynamic profile for a fast update filter with the specified name, and uses that filter.

If two different dynamic profiles include a fast update filter with the same name, the `match-order` specification of the two filters must be identical. If the two filters are activated on the same interface, the terms are added together.

The router includes the filter name in `show firewall` command results. The router also creates unique names for filter terms and counters for the `show firewall` command.

When a fast update filter is created by the activation of a dynamic profile, the router creates an interface-specific name for the filter. The name uses the following format, which is also used for classic filters:

<filter-name>-<interface-name>.<subunit>-<direction>

For example, an input filter named `httpFilter` on interface `ge-1/0/0.5` is named as follows (`in` indicates an input filter and `out` indicates an output filter):

`http-filter-ge-1/0/0.5-in`

The router creates unique names for the filter terms and counters by appending the session ID to all term and counter names. Terms that use the `only-at-create` statement have a session-id of 0. Terms and counters use the following format:

<term-name>-<session-id>

<counter-name>-<session-id>

Guidelines for Creating and Applying Fast Update Filters

Fast update filters enable you to create subscriber-specific firewall filters and dynamically apply these filters to statically created interfaces using dynamic profiles. Individual terms can be added to, or removed from, a filter without requiring that the entire filter be recompiled.

When creating and applying fast update filters, keep the following in mind:

- Dynamic application of input and output filters is supported.
- You cannot use the same fast update filter as both an input and output filter in the same dynamic profile attached to an interface.
- Fast update filters must always include terms that permit DHCP traffic to pass. See "Configuring Filters to Permit Expected Traffic" on page 359.
- You can create `family inet` and `inet6` filters.
- You can add or remove both IPv4 and IPv6 filters with the same service activation or deactivation.
- You can remove one filter type without impacting the other type of filter. For example, you can remove IPv6 filters and leave the current IPv4 filters active.

- The `interface-specific` statement is required for all fast update filters.
- The `match-order` statement is required—you must explicitly state the order of the match fields in a fast update filter. See "Configuring the Match Order for Fast Update Filters" on page 354.
- The `match-order` statement uses an implied wildcard for conditions that you specify in the statement. If you specify a condition that is not also configured in the `from` specification of a filter term, the router considers that a wildcard for that condition.
- A filter term can have only a single value or range; however, you can configure multiple terms to specify multiple ranges.
- You can match a maximum of five match conditions in a filter.

RELATED DOCUMENTATION

Fast Update Filter Actions and Action Modifiers | 356

Fast Update Filter Match Conditions | 355

Avoiding Conflicts When Terms Match | 360

Understanding Dynamic Firewall Filters | 236

Classic Filters Overview | 239

Dynamically Attaching Statically Created Filters for Any Interface Type | 256

Dynamically Attaching Statically Created Filters for a Specific Interface Family Type | 255

Verifying and Managing Firewall Filter Configuration | 426

Basic Fast Update Filter Syntax

This section shows the basic fast update filter statement syntax. The first part of this syntax provides the CLI statements to associate an input and output filter with a dynamic profile. The second part of this syntax represents the configured input and output filters associated to the dynamic profile. When a DHCP event occurs, the dynamic profile applies the specified filters to the DHCP client interface on the router.

```
[edit dynamic-profiles profile-name]
interfaces {
    $junos-interface-ifd-name {
        unit $junos-underlying-interface-unit {
            family family {
```

```
                filter {
                    input filter-name;
                    precedence precedence;
                    output filter-name;
                    precedence precedence;
                }
            }
        }
    }
}
[edit dynamic-profiles profile-name]
firewall {
    family family {
        fast-update-filter filter-name {
            [desired filter configuration]
        }
        fast-update-filter filter-name {
            [desired filter configuration]
        }
    }
}
```

RELATED DOCUMENTATION

Configuring Fast Update Filters | 349

Configuring Fast Update Filters

You configure a fast update filter in a dynamic profile—this enables you to use dynamic variables in the filter configuration. After you configure fast update filters, you then use the `dynamic-profiles` syntax to associate the filter with the subscriber interface.

To configure a fast update filter for subscriber access:

1. Access the dynamic profile you want to use.

```
[edit]
user@host# edit dynamic-profiles myProfile
```

2. Specify that you want to configure a firewall, and specify the family.

```
[edit dynamic-profiles myProfile]
user@host# edit firewall family inet
```

3. Specify that you want to configure a fast update filter and assign a name to the filter.

```
[edit dynamic-profiles myProfile firewall family inet]
user@host# edit fast-update-filter httpFilter
```

4. Specify the `interface-specific` statement. This statement is mandatory.

```
[edit dynamic-profiles myProfile firewall family inet fast-update-filter httpFilter]
user@host# set interface-specific
```

5. Configure the match order to use for the filter terms.

```
[edit dynamic-profiles myProfile firewall family inet fast-update-filter httpFilter]
user@host# set match-order [source-address protocol destination-port]
```

See "Configuring the Match Order for Fast Update Filters" on page 354.

6. Specify that you want to configure a term for the filter and assign the name to the term. Configure the match conditions and actions for the term.

```
[edit dynamic-profiles myProfile firewall family inet fast-update-filter httpFilter]
user@host# edit term term1

[edit dynamic-profiles myProfile firewall family inet fast-update-filter httpFilter term
term1]
user@host# set from protocol tcp
user@host# set from source-address $junos-subscriber-ip-address
user@host# set from destination-port http
user@host# set then count http-cnt
```

See "Configuring Terms for Fast Update Filters" on page 357.

RELATED DOCUMENTATION

- Configuring the Match Order for Fast Update Filters | 354
- Configuring Terms for Fast Update Filters | 357
- Associating Fast Update Filters with Interfaces in a Dynamic Profile | 367
- Fast Update Filters Overview | 344
- *Dynamic Profiles Overview*
- Guidelines for Configuring Firewall Filters
- Guidelines for Applying Standard Firewall Filters

Example: Configuring Fast Update Filters for Subscriber Access

This example shows you how to configure a fast update filter that is an input filter that counts the HTTP and non-HTTP packets from a subscriber. In the example, you use the firewall stanza to create the filter and the interfaces stanza to attach the filter.

```
[edit dynamic-profiles myProfile]
firewall {
    family inet {
        fast-update-filter httpFilter {
            interface-specific;
            match-order [source-address protocol destination-port];
            term term1 {
                from {
                    protocol tcp;
                    source-address $junos-subscriber-ip-address;
                    destination-port http;
                }
                then {
                    count http-cnt;
                }
            }
            term term2 {
                from {
                    protocol tcp;
                    source-address $junos-subscriber-ip-address;
                }
                then {
                    count non-http-cnt;
```

```
                }
            }
        }
    }
}
interfaces {
    "$junos-interface-ifd-name" {
        unit "$junos-underlying-interface-unit" {
            family inet {
                filter {
                    input httpFilter;
                }
            }
        }
    }
}
```

RELATED DOCUMENTATION

- Configuring Fast Update Filters | 349

Match Conditions and Actions in Fast Update Filters

IN THIS SECTION

- Match Conditions | 353
- Actions | 353
- Adding Terms Only Once | 353

To create a fast update filter, you use the `term` statement to specify conditions that a packet must have, and to specify the action the router performs when those conditions exist in the packet.

This section covers:

Match Conditions

Match conditions specify characteristics that a packet must have—if the conditions exist in the packet, the router then performs the specified action. You use the `from` keyword in the `term` statement to specify match conditions for the filter. The packet must match all conditions in the `from` specification for the action to be performed, which also means that their order in the `from` specification is not important.

An individual condition in a `from` specification can contain a single value or range. You can match a maximum of five match conditions in a filter.

"Fast Update Filter Match Conditions" on page 355 lists the match conditions you can use in fast update filters.

NOTE: The router uses an implied wildcard for conditions that you include in the `match-order` statement. If you include a condition that is *not* configured in the `from` specification of a filter term, the router considers that a wildcard for the condition.

For example, if you include the `dscp` condition in the `match-order` statement, but do not configure a `dscp` value in the `from` specification of the filter term, the router performs the action configured in the `then` specification of the filter on all DSCP values.

Actions

Actions and action modifiers specify the operation the router performs when a particular match condition exists in a packet. You use the `then` keyword in the `term` statement to specify the actions to perform on packets whose characteristics match the conditions specified in the preceding `from` specification.

Action modifiers are actions taken in addition to the specified action. You can configure any combination of action modifiers. For the action or action modifier to take effect, all conditions in the `from` specification must match. If you specify `log` as one of the actions in a term, this constitutes a termination action; whether any additional terms in the filter are processed depends on the traffic through the filter. The action modifier operations carry a default `accept` action. For example, if you specify an action modifier and do not specify an action, the specified action modifier is implemented and the packet is accepted.

"Fast Update Filter Actions and Action Modifiers" on page 356 lists the actions and action modifiers you can use in fast update filters.

Adding Terms Only Once

You can optionally specify that a term can be added only when the fast update filter is first created, and cannot be later changed by adding or removing conditions. We recommend that you only use the `only-`

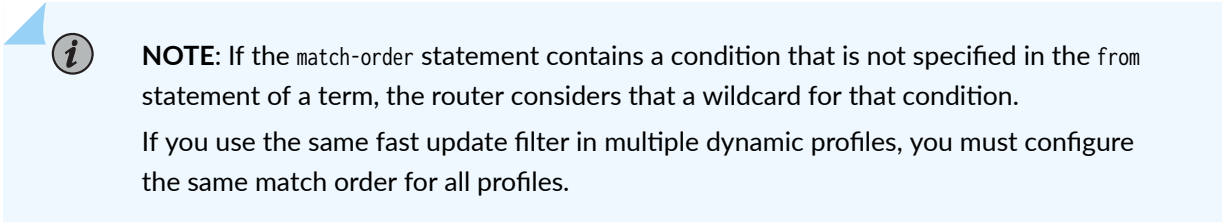
at-create option for terms that do not include subscriber-specific data in their match conditions, such as common or default terms (counting the default drop packet, for instance).

RELATED DOCUMENTATION

- Configuring Terms for Fast Update Filters | 357
- Fast Update Filter Match Conditions | 355
- Fast Update Filter Actions and Action Modifiers | 356

Configuring the Match Order for Fast Update Filters

You must include the match-order statement to explicitly specify the order in which router examines the match conditions. The router examines only those match conditions that you include in the statement. You can match a maximum of five conditions.



NOTE: If the match-order statement contains a condition that is not specified in the from statement of a term, the router considers that a wildcard for that condition.

If you use the same fast update filter in multiple dynamic profiles, you must configure the same match order for all profiles.

To configure the order in which the router examines the match conditions of a fast update filter:

1. Access the fast update filter:

```
[edit dynamic-profiles myProfile]
user@host# edit firewall family inet fast-update-filter httpFilter
```

2. Specify the mandatory interface-specific statement.

```
[edit dynamic-profiles myProfile firewall family inet fast-update-filter httpFilter]
user@host# set interface-specific
```

3. Configure the match order for the match conditions in the filter. Use brackets to enclose multiple match conditions.

```
[edit dynamic-profiles myProfile firewall family inet fast-update-filter httpFilter]
user@host# set match-order [source-address protocol destination-port]
```

RELATED DOCUMENTATION

- Configuring Fast Update Filters | 349
- Configuring Terms for Fast Update Filters | 357
- Fast Update Filters Overview | 344
- *Dynamic Profiles Overview*
- Fast Update Filter Match Conditions | 355
- Guidelines for Configuring Firewall Filters

Fast Update Filter Match Conditions

Table 32: Fast Update Filter Match Conditions

Match Condition	Description
`destination-address` *prefix*	IP destination address field.
`destination-port` *number*	TCP or UDP destination port field. Can be a single number, a single range, or one of the standard port synonyms.
`dscp` *number*	Differentiated services code point. Can be a single number, a single range, or the standard synonyms. IPv4 only.
`match-terms` *string-of-conditions*	Series of match conditions. Enclose the string within quotation marks and use semicolons to separate entries. For example, `match-terms “protocol tcp; destination-port http”;`. Dynamic profile variables are not allowed in the string.

Table 32: Fast Update Filter Match Conditions *(Continued)*

Match Condition	Description
`protocol` *number*	IP protocol field. Can be a single number, a single range, or one of the standard protocol synonyms. IPv4 only.
`source-address` *prefix*	IP source address field.
`source-port` *number*	TCP or UDP source port field. Can be a single number, a single range, or one of the standard protocol synonyms.

RELATED DOCUMENTATION

Configuring Fast Update Filters | 349

Fast Update Filter Actions and Action Modifiers

Table 33: Fast Update Filter Actions and Action Modifiers

Action or Action Modifier	Description
Actions	
`accept`	Accept the packet.
`action-terms` *string-of-actions*	A series of multiple actions or action modifiers. Enclose the string within quotation marks and use semicolons to separate entries. For example, `action-terms "log; count http-cnt";`. Dynamic profile variables are not allowed in the string.
`discard`	Drop the packet silently, without sending an Internet Control Message Protocol (ICMP) message.

Table 33: Fast Update Filter Actions and Action Modifiers *(Continued)*

Action or Action Modifier	Description
`ignore-term`	Do not add this term to the filter. All match conditions and actions are ignored.
`port-mirror`	Port mirror packets.
`routing-instance` *routing-instance*	Forward packets to specified routing instance.
Action Modifiers	
`count` *counter-name*	Increment the specified counter.
`forwarding-class` *class*	Classify the packet into one of the following forwarding classes: `as`, `assured-forwarding`, `best-effort`, `expedited-forwarding`, or `network-control`.
`log`	Log the packet header information.
`loss-priority (high | medium-high | medium-low| low)`	Set the loss priority level for packets.
`policer` *policer-name*	Rate-limit packets based on the specified policer.

RELATED DOCUMENTATION

Configuring Fast Update Filters | 349

Configuring Terms for Fast Update Filters

A fast update filter consists of one or more terms. A term is made up of one or more match conditions and the action to take when a packet matches the specified conditions.

To configure a term for a fast update filter:

1. Access the fast update filter.

```
[edit dynamic-profiles myProfile]
user@host# edit firewall family inet fast-update-filter httpFilter
```

2. Create the new term and assign a name to the term.

```
[edit dynamic-profiles myProfile firewall family inet fast-update-filter httpFilter]
user@host# set term term1
```

3. Configure the match condition for the term. See "Fast Update Filter Match Conditions" on page 355 for the supported match conditions for fast update filters.

```
[edit dynamic-profiles myProfile firewall family inet fast-update-filter httpFilter]
user@host# set from protocol tcp
user@host# set from source-address $junos-subscriber-ip-address
user@host# set from destination-port http
```

4. Configure the action that the router takes when the match conditions are met. See "Fast Update Filter Actions and Action Modifiers" on page 356 for the supported actions for fast update filters.

```
[edit dynamic-profiles myProfile firewall family inet fast-update-filter httpFilter]
user@host# set then accept
```

5. (Optional) Configure the action modifiers that you want the router to take when the match conditions are met. See "Fast Update Filter Actions and Action Modifiers" on page 356 for the supported action-modifiers for fast update filters.

```
[edit dynamic-profiles myProfile firewall family inet fast-update-filter httpFilter]
user@host# set then count http-cnt
```

6. (Optional) Configure the term to be added only once, when the fast update filter is first created.

```
[edit dynamic-profiles myProfile firewall family inet fast-update-filter httpFilter]
user@host# set only-at-create
```

RELATED DOCUMENTATION

- Configuring Fast Update Filters | 349
- Configuring the Match Order for Fast Update Filters | 354
- Fast Update Filters Overview | 344
- Fast Update Filter Match Conditions | 355
- Fast Update Filter Actions and Action Modifiers | 356
- Stateless Firewall Filter Overview
- Stateless Firewall Filter Components

Configuring Filters to Permit Expected Traffic

You must explicitly configure your *firewall filter* to permit expected traffic, such as DHCP traffic, to pass. Otherwise, the expected traffic is denied when the filter is applied to the interface. This requirement applies to both classic and fast update filters.

The following example shows a fast update filter that might be used to accept DHCP traffic. The actual filter you use depends on the expected traffic in your network.

In the example, the term `allow-dhcp` accepts all DHCP traffic from all source addresses. The term also includes the `only-at-create` option to specify that the term is applied only when the filter is first applied. The term `sub-allow-dhcp` includes the Junos OS predefined variable `$junos-subscriber-ip-address`, which permits all subscriber-specific DHCP traffic.

The `match-order` statement configuration lists the conditions from most-specific to least-specific, as recommended in "Configuring the Match Order for Fast Update Filters" on page 354. Because this filter is designed to permit ingress DHCP traffic, the `source-address` condition is listed first.

```
firewall {
    family inet {
        fast-update-filter psf1 {
            interface-specific;
            match-order [ source-address destination-address protocol destination-port ];
            term allow-dhcp {
                only-at-create;
                from {
                    source-address 0.0.0.0/32;
                    destination-address 255.255.255.255/32;
                    destination-port 67;
                    protocol udp;
```

```
                }
                then accept;
            }
            term sub-allow-dhcp {
                from {
                    source-address $junos-subscriber-ip-address;
                    destination-address 192.168.1.2/32;
                    destination-port 67;
                    protocol udp;
                }
                then accept;
            }
        }
    }
}
```

RELATED DOCUMENTATION

Configuring the Match Order for Fast Update Filters | 354

Configuring Terms for Fast Update Filters | 357

Avoiding Conflicts When Terms Match

IN THIS SECTION

- How the Router Evaluates Terms in a Filter | 361
- Using Implied Wildcards | 362
- Conflict Caused by Overlapping Ranges | 364

A fast update filter can contain multiple terms, each with a variety of match conditions. However, when you configure multiple terms in a filter, you must ensure that the terms do not overlap, or conflict with each other. Two terms are considered to overlap when it is possible for a packet to match all conditions of both terms. Because each term specifies a different action for matches, the router cannot determine which action to take. When terms overlap, a conflict error occurs and the session fails when the dynamic profile attempts to apply the filter. The error log indicates the overlapping terms.

How the Router Evaluates Terms in a Filter

The router creates a table of match conditions when examining terms. The table, which is similar to a routing table, is based on the conditions included in the `match-order` statement. When the router receives a packet, the router examines the packet's contents in the sequence specified in the `match-order` statement.

For example, using the sample configuration in the following Match-Order Example, the router first examines the packet's `source-address`, then the `destination-address`, and finally the `destination-port`. As shown in the following table, the two terms in the filter do not overlap because each term has a different `destination-port` specification. The router then takes the appropriate filter action for the term that matches the `destination-port` value of the packet.

Term	source-address	destination-address	destination-port	Action
t55	subscriber's address	203.0.113.2/32	http	count t55_cntr accept
t999	subscriber's address	203.0.113.2/32	https	count t999_cntr accept

Match-Order Example

```
firewall {
    family inet {
        fast-update-filter psf1 {
            interface-specific;
            match-order [ source-address destination-address destination-port ];
            term t55 {
                from {
                    source-address $junos-subscriber-ip-address;
                    destination-address 203.0.113.2/32;
                    destination-port http;
                }
                then {
                    count t55_cntr;
                    accept;
                }
            }
            term t999 {
```

```
                from {
                    source-address $junos-subscriber-ip-address;
                    destination-address 203.0.113.2/32;
                    destination-port https;
                }
                then {
                    count t999_cntr;
                    accept;
                }
            }
        }
    }
}
```

Using Implied Wildcards

This section shows an example of how you might use an implied wildcard specification in the match configuration. A condition in the `match-order` statement is an implied wildcard when that condition is not configured in the `from` specification of a term in the filter.

NOTE: When you use ranges (for example, a range of values or a wildcard) in terms, the ranges must not overlap—overlapping ranges create a conflict error. However, you can configure a range in one term and an exact match in another term. For example, in the following filter table, the wildcard destination port value in term `t3` does not overlap the destination port specifications in terms `t55` and `t999` because the `http` and `https` values are exact matches.

In the Implied Wildcard Example configuration, the router views the `destination-port` condition in the `match-order` statement as an implied wildcard for term `t3`, because there is no `destination-port` value configured in that term. As a result, the wildcard specifies that for term `t3` any `destination-port` value is accepted. The filter table appears as follows:

Term	source-address	destination-address	destination-port	Action
t3	subscriber's address	203.0.113.2/32	any (wildcard)	count t3_cntr accept

(Continued)

Term	source-address	destination-address	destination-port	Action
t55	subscriber's address	203.0.113.2/32	http	count t55_cntr accept
t999	subscriber's address	203.0.113.2/32	https	count t999_cntr accept

In the following filter configuration, traffic with a destination port of `http` matches term `t55` and traffic with a destination port of `https` matches term `t999`. Traffic with a destination port other than `http` or `https` matches term `t3`, which is the implied wildcard.

Implied Wildcard Example

```
firewall {
    family inet {
        fast-update-filter psf1 {
            interface-specific;
            match-order [ source-address destination-address dscp protocol destination-port ];
            term t3 {
                from {
                    source-address $junos-subscriber-ip-address;
                    destination-address 203.0.113.2/32;
                }
                then {
                    count t3_cntr;
                    accept;
                }
            }
            term t55 {
                from {
                    source-address $junos-subscriber-ip-address;
                    destination-address 203.0.113.2/32;
                    destination-port http;
                }
                then {
                    count t55_cntr;
                    accept;
```

```
                }
            }
            term t999 {
                from {
                    source-address $junos-subscriber-ip-address;
                    destination-address 203.0.113.2/32;
                    destination-port https;
                }
                then {
                    count t999_cntr;
                    accept;
                }
            }
        }
    }
}
```

Conflict Caused by Overlapping Ranges

This section shows two examples of overlapping ranges in terms. When you use ranges (such as a wildcard or a range of values) in terms, the ranges must not overlap—overlapping ranges create a conflict error and the session fails.

In the following filter configuration, the `destination-port` ranges in the two terms overlap. Ports in the range from 50 through 80 match both term `src0` and term `src1`, which each specify different actions to take.

NOTE: You can configure a range in one term and an exact match in another term. See the section, *Using Implied Wildcards*, for an example that uses a wildcard for a match condition in one term and an exact match for the condition in a second term.

Term	source-address	destination-address	destination-port	Action
src0	subscriber's address	203.0.113.2/32	0–80	count c1_cntr accept
src1	subscriber's address	203.0.113.2/32	50–100	count c2_cntr accept

Overlapping Ranges Example 1

```
firewall {
    family inet {
        fast-update-filter fuf-src {
            interface-specific;
            match-order [ source-address destination-address destination-port ];
            term src0 {
                from {
                    source-address $junos-subscriber-ip-address;
                    destination-address 203.0.113.2/32;
                    destination-port 0-80;
                }
                then {
                    count c1_cntr;
                    accept;
                }
            }
            term src1 {
                from {
                    source-address $junos-subscriber-ip-address;
                    destination-address 203.0.113.2/32;
                    destination-port 50-100;
                }
                then {
                    count c2_cntr;
                    accept;
                }
            }
        }
```

In this filter configuration, the `protocol` specification in terms `src21` and `src22` use the implied wildcard, which configures a range for each term. Because overlapping ranges are not allowed, a conflict error results.

Term	source-address	destination-address	protocol	destination-port	Action
src20	subscriber's address	203.0.113.2/32	udp	any (wildcard)	count c20_cntr accept

(Continued)

Term	source-address	destination-address	protocol	destination-port	Action
src21	subscriber's address	203.0.113.2/32	any (wildcard)	http	count c21_cntr accept
src21	subscriber's address	203.0.113.2/32	any (wildcard)	https	count c22_cntr accept

Overlapping Ranges Example 2

```
firewall {
    family inet {
        fast-update-filter fuf-src2 {
            interface-specific;
            match-order [ source-address destination-address protocol destination-port ];
            term src20 {
                from {
                    source-address $junos-subscriber-ip-address;
                    destination-address 203.0.113.2/32;
                    protocol udp;
                }
                then {
                    count c20_cntr;
                    accept;
                }
            }
            term src21 {
                from {
                    source-address $junos-subscriber-ip-address;
                    destination-address 203.0.113.2/32;
                    destination-port http;
                }
                then {
                    count c21_cntr;
                    accept;
                }
            }
```

```
            term src22 {
                from {
                    source-address $junos-subscriber-ip-address;
                    destination-address 203.0.113.2/32;
                    destination-port https;
                }
                then {
                    count c22_cntr;
                    accept;
                }
            }
        }
```

RELATED DOCUMENTATION

- Configuring Fast Update Filters | 349
- Configuring Terms for Fast Update Filters | 357
- Configuring the Match Order for Fast Update Filters | 354

Associating Fast Update Filters with Interfaces in a Dynamic Profile

After you configure the fast update filter, you reference the filter in the `interfaces` stanza of a dynamic profile. When the dynamic profile instantiates a subscriber session, the router applies the terms of the filter to the interface.

To apply a fast update filter to an interface in a dynamic profile:

1. Access the dynamic profile you want to use.

```
[edit]
user@host# edit dynamic-profiles myProfile
```

2. Specify the interface for the dynamic profile—use the dynamic interface variable.

```
[edit dynamic-profiles myProfile]
user@host# edit interfaces $junos-interface-ifd-name
```

3. Specify the underlying interface—use the unit number variable.

```
[edit dynamic-profiles myProfile interfaces “$junos-interface-ifd-name”]
user@host# edit unit $junos-underlying-interface-unit
```

4. Specify the family. Use `inet` if you are using IPv4 filters or `inet6` for IPv6 filters.

```
[edit dynamic-profiles myProfile interfaces “$junos-interface-ifd-name” unit “$junos-
underlying-interface-unit”]
user@host# edit family inet
```

5. Specify the filters that you want to apply to the interface.

```
[edit dynamic-profiles myProfile interfaces “$junos-interface-ifd-name” unit “$junos-
underlying-interface-unit” family inet]
user@host# set filter input httpFilter
user@host# set filter output myOutFilter
```

RELATED DOCUMENTATION

Dynamic Profiles Overview

Fast Update Filters Overview | 344

Guidelines for Configuring Firewall Filters

Guidelines for Applying Standard Firewall Filters

CHAPTER 22

Defending Against DoS and DDoS Attacks Using Unicast RPF and Fail Filters

IN THIS CHAPTER

- Unicast RPF in Dynamic Profiles for Subscriber Interfaces | 369

Unicast RPF in Dynamic Profiles for Subscriber Interfaces

IN THIS SECTION

- Unicast RPF in Dynamic Profiles for Subscriber Interfaces | 369
- Configuring Unicast RPF in Dynamic Profiles for Subscriber Interfaces | 370
- Configuring a Fail Filter for Unicast RPF in Dynamic Profiles for Subscriber Interfaces | 371
- Example: Configuring Unicast RPF in a Dynamic Profile on MX Series Routers | 372

Unicast RPF in Dynamic Profiles for Subscriber Interfaces

Unicast reverse-path forwarding (RPF) provides a way to reduce the effect of denial-of-service (DoS) and distributed denial-of-service (DDoS) attacks on IPv4 and IPv6 interfaces. When you configure unicast RPF on an interface, it checks the packet source address. Packets that pass the check are forwarded. Packets that fail the check are dropped, or if a fail filter is configured, are passed to the filter for further evaluation.

Unicast RPF has two behavioral modes, *strict* and *loose*. When you configure unicast RPF in a dynamic profile, strict mode is the default. In strict mode, unicast RPF checks whether the source address of the incoming packet matches a prefix in the routing table, and whether the interface expects to receive a packet with this source address prefix. In loose mode, unicast RPF checks only whether the source address has a match in the routing table. It does not check whether the interface expects to receive a packet from a specific source address.

For both modes, when an incoming packet fails the unicast RPF check, the packet is not accepted on the interface. Instead, unicast RPF counts the packet and sends it to an optional fail filter, if present. The fail filter determines what further action is taken on the packet. In the absence of a fail filter, the packet is silently discarded.

NOTE:

- Packet counter is not supported on Junos OS Evolved.
- On Junos OS Evolved, you do not need a fail-filter because the system prevents dropping of Bootstrap Protocol (BOOTP) packets and Dynamic Host Configuration Protocol (DHCP) packets during the RPF check.

The `show interfaces statistics` *`logical-interface-name`* `detail` command displays unicast RPF statistics for dynamic logical interfaces when either `rpf-check` or `rpf-check mode loose` is enabled on the interface. No additional statistics are displayed when `rpf-check fail-filter` *`filter-name`* is configured on the interface. The `clear interfaces statistics` *`logical-interface-name`* command clears RPF statistics.

Configuring Unicast RPF in Dynamic Profiles for Subscriber Interfaces

Unicast RPF provides a way to reduce the effect of denial-of-service attacks on IPv4 and IPv6 interfaces by checking the source IP address against the routing table. Packets that do not match are silently discarded, unless an optional fail filter is configured. The fail filter performs an additional check and directs some action be taken on certain packets. Typical actions include logging the packets or passing them even though they failed the RPF check.

NOTE: Although the fail filter is technically optional, for dynamic profiles in a DHCP environment you must configure a filter to pass DHCP packets. By default, the RPF check prevents DHCP packets from being accepted on interfaces protected by the RPF check. The fail filter identifies the DHCP packets and passes them on.

To configure a unicast RPF check in a dynamic profile:

1. Access the dynamic profile.

```
[edit]
user@host# edit  dynamic-profiles profile-name
```

2. Access the interface and specify the address family

```
[edit dynamic-profiles profile-name]
user@host# edit interfaces interface-name unit logical-unit-number family inet
```

> **NOTE**: On Junos OS, you must specify address families (`inet` and `inet6`) separately for IPv4 and IPv6. On Junos OS Evolved, you only need to specify either inet or inet6 to enable RPF check for both families.

3. Enable the RPF check in strict or loose mode.

   - Configure strict mode to check whether the source address of the incoming packet matches a prefix in the routing table, and whether the interface expects to receive a packet with this source address prefix:

     ```
     [edit dynamic-profiles profile-name interface interface-name unit logical-unit-number
     family inet]
     user@host# set rpf-check
     ```

   - Configure loose mode to check only whether the source address has a match in the routing table:

     ```
     [edit dynamic-profiles profile-name interface interface-name unit logical-unit-number
     family inet]
     user@host# set rpf-check mode loose
     ```

4. (Optional except for DHCP) Enable the RPF check and specify the fail filter.

   ```
   [edit dynamic-profiles profile-name interface interface-name unit logical-unit-number family
   inet]
   user@host# set rpf-check fail-filter filter-name
   ```

   For information about defining a fail filter, see "Configuring a Fail Filter for Unicast RPF in Dynamic Profiles for Subscriber Interfaces" on page 371.

Configuring a Fail Filter for Unicast RPF in Dynamic Profiles for Subscriber Interfaces

This topic describes how to configure a fail filter at the `[edit firewall]` hierarchy level that can be optionally applied by unicast RPF for subscriber interfaces in dynamic profiles on MX Series routers.

NOTE: In contrast to statically configured fail filters, RPF-check fail filters used in a dynamic profile cannot be specific to a particular interface.

To configure a firewall fail filter:

1. Create the filter.

```
[edit]
user@host# edit firewall family inet filter filter-name
```

2. Specify a term for the filter.

```
[edit firewall family inet filter filter-name]
user@host# edit term term-name
```

3. Configure the match conditions for the filter.

```
[edit firewall family inet filter filter-name term term-name]
user@host# set  from match-conditions
```

4. Configure the actions to be taken for the matching packets.

```
[edit firewall family inet filter filter-name term term-name]
user@host# set  then actions
```

5. (Optional) Repeat Steps 3 and 4 for additional filter terms.

Example: Configuring Unicast RPF in a Dynamic Profile on MX Series Routers

IN THIS SECTION

- Requirements | 373
- Overview | 373
- Configuration | 374
- Verification | 379

This example shows how to help defend the router ingress interfaces against denial-of-service (DoS) and distributed denial-of-service (DDoS) attacks by configuring unicast reverse-path forwarding (RPF) on a customer-edge interface to filter incoming traffic. Unicast RPF verifies the unicast source address of each packet that arrives on an ingress interface where unicast RPF is enabled. Packets that fail verification are silently discarded unless a fail filter performs some other action on them.

Requirements

This example uses the following software and hardware components:

- An MX Series 5G Universal Routing Platform

Before you begin:

- Configure the dynamic profile that you intend to use to apply the RPF check.

  See *Configuring a Basic Dynamic Profile*.

Overview

IN THIS SECTION

- Topology | **374**

Large amounts of unauthorized traffic—such as attempts to flood a network with fake service requests in a denial-of-service (DoS) attack—can consume network resources and deny service to legitimate users. One way to help prevent DoS and distributed denial-of-service (DDoS) attacks is to verify that incoming traffic originates from legitimate network sources.

Unicast RPF helps ensure that a traffic source is legitimate (authorized) by comparing the source address of each packet that arrives on an interface to the forwarding-table entry for its source address. If the router uses the same interface that the packet arrived on to reply to the packet's source, this verifies that the packet originated from an authorized source, and the router forwards the packet. If the router does not use the same interface that the packet arrived on to reply to the packet's source, the packet might have originated from an unauthorized source, and the router discards the packet, or passes it to a fail filter.

The fail filter enables you to set criteria for packets you want to be passed in spite of failing the RPF check, such as DHCP packets, which are dropped by default.

On MX Series routers, you can configure unicast RPF in a dynamic profile to apply the configuration to one or more subscriber interfaces. See Understanding Unicast RPF (Routers) for more information about the behavior and limitations of unicast RPF on MX Series routers.

In this example, you configure the router to protect against potential DoS and DDoS attacks from the Internet perpetrated through IPv4 packets arriving on dynamically created VLAN demux interfaces. The dynamic profile, vlan-demux-prof, establishes that VLAN demux interfaces are automatically created for subscribers. Unicast RPF is enabled on the dynamic interfaces by the rpf-check term.

By default, unicast RPF prevents Dynamic Host Configuration Protocol (DHCP) packets from being accepted on interfaces to which it applies. When DHCP packets are discarded, no new subscribers can be created by the dynamic profile. To enable interfaces to accept DHCP packets, you must apply a fail filter that properly sorts through the packets that fail the check and identifies the DHCP packets. In this example, you configure the `allow-dhcp` term in the filter `rpf-pass-dhcp`. This term matches, counts, and accepts IPv4 packets that are destined for the DHCP port and any address. The `default term` drops all other packets that fail the RPF check.

This example does not show all possible configuration choices.

Topology

Configuration

IN THIS SECTION

- Configuring the Dynamic Profile to Apply RPF Checking to Dynamic VLAN Demux Interfaces | **375**
- Configuring the RPF-Check Fail Filter | **376**

To enable unicast RPF with a fail filter in a dynamic profile, perform these tasks:

Configuring the Dynamic Profile to Apply RPF Checking to Dynamic VLAN Demux Interfaces

CLI Quick Configuration

To quickly configure the dynamic profile to apply unicast RPF to dynamically created VLAN demux interfaces, copy the following commands, paste them in a text file, remove any line breaks, and then copy and paste the commands into the CLI.

```
edit dynamic-profiles vlan-demux-prof interfaces demux0
edit unit $junos-interface-unit
set demux-options underlying-interface $junos-interface-ifd-name
set vlan-id $junos-vlan-id
edit family inet
set unnumbered-address lo0.0
set rpf-check fail-filter rpf-pass-dhcp
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure unicast RPF on the router:

1. Create a dynamic profile.

```
[edit]
user@host# edit dynamic-profiles vlan-demux-prof
```

2. Specify that the dynamic VLAN profile use the demux interface.

```
[edit dynamic-profiles vlan-demux-prof]
user@host# edit interfaces  demux0
```

3. Specify that the dynamic profile applies the demux interface unit value to the dynamic VLANs.

```
[edit dynamic-profiles vlan-demux-prof interfaces demux0]
user@host# edit unit $junos-interface-unit
```

4. Specify the logical underlying interface for the dynamic VLANs.

```
[edit dynamic-profiles vlan-demux-prof interfaces demux0 unit $junos-interface-unit]
user@host# set demux-options underlying-interface $junos-interface-ifd-name
```

5. Configure the variable that results in dynamically created VLAN IDs.

```
[edit dynamic-profiles vlan-demux-prof interfaces demux0 unit $junos-interface-unit]
user@host# set vlan-id $junos-vlan-id
```

6. Configure the IPv4 address family for the demux interfaces.

```
[edit dynamic-profiles vlan-demux-prof interfaces demux0 unit $junos-interface-unit]
user@host# edit family inet
```

7. Configure the unnumbered address for the family.

```
[edit dynamic-profiles vlan-demux-prof interfaces demux0 unit $junos-interface-unit family
inet]
user@host# set unnumbered-address lo0.0
```

8. Configure unicast RPF and specify the fail filter that is applied to incoming packets that fail the check.

```
[edit dynamic-profiles vlan-demux-prof interfaces demux0 unit $junos-interface-unit family
inet]
user@host# set fail-filter fail-filter rpf-pass-dhcp
```

Configuring the RPF-Check Fail Filter

CLI Quick Configuration

To quickly configure the unicast RPF-check fail filter, copy the following commands, paste them in a text file, remove any line breaks, and then copy and paste the commands into the CLI.

```
edit firewall family inet filter rpf-pass-dhcp
edit term allow-dhcp
```

```
set from destination-port dhcp
set from destination-address 255.255.255.255/32
set then count rpf-dhcp-traffic
set then accept
up
edit term default
set then discard
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure the RPF-check fail filter:

1. Create the fail filter.

```
[edit firewall]
user@host# edit family inet filter rpf-pass-dhcp
```

2. Define the filter term that identifies DHCP packets based on the DHCP destination port, then counts and passes the packets.

```
[edit firewall family inet filter rpf-pass-dhcp]
user@host# edit term allow-dhcp
user@host# set  from destination-port dhcp
user@host# set  from destination-address 255.255.255.255/32
user@host# set  then count rpf-dhcp-traffic
user@host# set  then accept
```

3. Define the filter term that drops all other failed packets.

```
[edit firewall filter rpf-pass-dhcp]
user@host# edit term default
user@host# set  then discard
```

Results

From configuration mode, confirm the unicast RPF configuration by entering the `show dynamic-profiles` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show dynamic-profiles
vlan-demux-prof {
    interfaces {
        demux0 {
            unit "$junos-interface-unit" {
                vlan-id "$junos-vlan-id";
                demux-options {
                    underlying-interface "$junos-interface-ifd-name";
                }
                family inet {
                    unnumbered-address lo0.0;
                    rpf-check {
                        fail-filter rpf-pass-dhcp;
                    }
                }
            }
        }
    }
}
```

From configuration mode, confirm the fail filter configuration by entering the `show firewall` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show firewall
family inet {
    filter rpf-pass-dhcp {
        term allow-dhcp {
            from {
                destination-address {
                    255.255.255.255/32;
                }
                destination-port dhcp;
```

```
            }
            then {
                count rpf-dhcp-traffic;
                accept;
            }
        }
        term default {
            then {
                discard;
            }
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Verification

IN THIS SECTION

- Verifying That Unicast RPF Is Enabled on the Router | **379**

To confirm that the configuration is correct, perform these tasks:

Verifying That Unicast RPF Is Enabled on the Router

Purpose

Verify that unicast RPF is enabled.

Action

Verify that unicast RPF is enabled by using the `show subscribers extensive` command.

```
user@host> show subscribers extensive
Type: VLAN
  Logical System: default
  Routing Instance: default
```

```
Interface: ae0.1073741824
Interface type: Dynamic
Dynamic Profile Name: vlan-demux-prof
State: Active
Session ID: 9
VLAN Id: 100
Login Time: 2011-08-26 08:17:00 PDT
IPv4 rpf-check Fail Filter Name: rpf-pass-dhcp
```

Meaning

The IPv4 rpf-check Fail Filter Name field displays `rpf-pass-dhcp`, the name of the fail filter applied by the dynamic profile for IPv4 packets failing the RPF check.

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
25.4R1	Starting in Junos OS Evolved Release 25.4R1, the Broadband Network Gateways (BNGs) running Junos OS Evolved support the unicast RPF filter.
19.1R1	Starting in Junos OS Release 19.1R1, the `show interfaces statistics` *`logical-interface-name`* `detail` command displays unicast RPF statistics for dynamic logical interfaces when either `rpf-check` or `rpf-check mode loose` is enabled on the interface.

RELATED DOCUMENTATION

- Understanding Unicast RPF (Routers)

CHAPTER 23

Improving Scaling and Performance of Filters on Static Subscriber Interfaces

IN THIS CHAPTER

- Firewall Filters and Enhanced Network Services Mode Overview | 381
- Configuring a Filter for Use with Enhanced Network Services Mode | 384

Firewall Filters and Enhanced Network Services Mode Overview

Under normal conditions, every *firewall filter* is generated in two different formats -- compiled and term-based. The compiled format is used by the routing engine (RE) kernel, FPCs, and MS-DPs. The term-based format is used by MPCs. Compiled firewall filters are duplicated for each interface or *logical interface* to which they are applied. Term-based filters, instead of being duplicated, are referenced by each interface or logical interface.

When a combination of MPCs and any other cards populate a chassis, the creation of both firewall filter file formats is necessary. In most networks, the creation of both filter formats and any amount of duplication for compiled firewall filters has no effect on the router. However, in subscriber management networks that include thousands of statically configured subscriber interfaces, creating filters in multiple formats and duplicating those filters for each interface can utilize a large portion of router memory resources. You can use either Enhanced IP Network Services mode or Enhanced Ethernet Network Services mode to improve the scaling and performance specific to routing filters in a subscriber access network that uses statically configured subscriber interfaces.

In configurations where interfaces are created either statically or dynamically and firewall filters are applied dynamically, you must configure the chassis network services to run in enhanced mode. In configurations where interfaces are created statically and firewall filters are applied statically, you must configure chassis network services to run in enhanced mode and also configure each firewall filter for enhanced mode.

NOTE: Do not use enhanced mode for firewall filters that are intended for control plane traffic. Control plane filtering is handled by the Routing Engine kernel, which cannot use the term-based format of the enhanced mode filters.

Table 34 on page 382 shows the configuration options when determining enhanced network services mode usage.

Table 34: Enhanced Network Services Mode and Firewall Filter Use Case Determination

Interface and Filter Configuration	Chassis Enhanced Mode Required	Firewall Filter Enhanced Mode Required
Dynamically-created interfaces and dynamically-applied filters	Yes	No
Statically-created interfaces and dynamically-applied filters	Yes	No
Statically-created interfaces and statically-applied filters	Yes	Yes

To achieve significant resource savings for the router, combine chassis and filter enhanced mode configuration as follows:

- Install only MPCs in the chassis.

  NOTE: Configuring chassis network services to run one of the enhanced network services modes results in the router enabling only MPCs and MS-DPCs. Because MS-DPCs use compiled firewall filter format, a router chassis that is configured for one of the enhanced network services modes, configuring standard (non-enhanced) firewall filters for use with any MS-DPCs can decrease optimal resource efficiency.

- When configuring static interfaces on the router, configure chassis network services to run either Enhanced IP Network Services mode or Enhanced Ethernet Network Services mode.
- When statically applying firewall filters to statically-created interfaces, configure any firewall filters for enhanced mode to limit the filter creation to only term-based format.

NOTE: Any firewall filters that are not configured for enhanced mode are created in both compiled and term-based format, even if the chassis is running one of the enhanced network services modes. Only term-based (enhanced) firewall filters will be generated, regardless of the setting of the `enhanced-mode` statement at the [`edit chassis network-services`] hierarchy level, if any of the following are true:

- Flexible filter match conditions are configured at the [`edit firewall family` *`family-name`* `filter` *`filter-name`* `term` *`term-name`* `from`] or [`edit firewall filter` *`filter-name`* `term` *`term-name`* `from`] hierarchy levels.
- A tunnel header push or pop action, such as GRE encapsulate or decapsulate is configured at the [`edit firewall family` *`family-name`* `filter` *`filter-name`* `term` *`term-name`* `then`] hierarchy level.
- Payload-protocol match conditions are configured at the [`edit firewall family` *`family-name`* `filter` *`filter-name`* `term` *`term-name`* `from`] or [`edit firewall filter` *`filter-name`* `term` *`term-name`* `from`] hierarchy levels.
- An extension-header match is configured at the [`edit firewall family` *`family-name`* `filter` *`filter-name`* `term` *`term-name`* `from`] or [`edit firewall filter` *`filter-name`* `term` *`term-name`* `from`] hierarchy levels.
- A match condition is configured that only works with MPC cards, such as firewall bridge filters for IPv6 traffic.

WARNING: Any firewall filter meeting the previous criteria will not be applied to the loopback, lo0, interface of DPC based FPCs. This means that term-based (enhanced) filters configured for use on the loopback interface of a DPC based FPC will not be applied. This will leave the RE unprotected by that filter.

RELATED DOCUMENTATION

Network Services Mode Overview

Configuring Junos OS to Run a Specific Network Services Mode in MX Series Routers

Configuring a Filter for Use with Enhanced Network Services Mode | 384

Configuring a Filter for Use with Enhanced Network Services Mode

For a statically-applied enhanced mode filter to function on statically created interfaces, you must include the `enhanced mode` statement in each filter. However, you do not need to configure the `enhanced mode` statement in filters that are dynamically applied to either static or dynamically-created interfaces.

NOTE: For either static or dynamic interfaces to use enhanced network services mode, you must configure the router chassis network services to use either Enhanced IP Network Services mode or Enhanced Ethernet Network Services mode. By configuring chassis network services to run in one of the enhanced modes, the router enables only MPCs and MS-DPCs in the chassis. See "Firewall Filters and Enhanced Network Services Mode Overview" on page 381 for details.

To configure a stateless firewall filter to use enhanced mode:

1. Create or edit the stateless firewall filter.

   NOTE: You can configure enhanced mode firewall filters for only `inet` and `inet6` filter families.

   For IPv4:

   ```
   [edit]
   user@host# edit firewall family inet filter filter-name
   ```

   For IPv6:

   ```
   [edit]
   user@host# edit firewall family inet6 filter filter-name
   ```

2. Specify the filter as an enhanced mode filter.

   ```
   [edit firewall family inet filter filter-name]
   user@host# set enhanced-mode
   ```

3. Configure or modify any filter terms.

   See Example: Configuring and Applying a Simple Filter for a filter configuration example.

RELATED DOCUMENTATION

- Understanding How to Use Standard Firewall Filters
- Network Services Mode Overview
- Firewall Filters and Enhanced Network Services Mode Overview | **381**
- Configuring Junos OS to Run a Specific Network Services Mode in MX Series Routers
- Understanding Dynamic Firewall Filters | **236**

CHAPTER 24

Configuring Dynamic Service Sets

IN THIS CHAPTER

- Dynamic Service Sets Overview | 386
- Associating Service Sets with Interfaces in a Dynamic Profile | 387
- Verifying and Managing Service Sets Information | 388

Dynamic Service Sets Overview

A service set is a collection of services to be performed by an Adaptive Services (AS) or Multiservices PIC. You configure a service-set definition at the `[edit services]` hierarchy level. You can then apply the service set to one or more interfaces on the router. The service set can be applied either dynamically or statically.

To dynamically associate a service set to interfaces you include the `service-set` statement with the `input` or `output` statement at the `[edit dynamic-profiles` *`profile-name`* `interfaces` *`interface-name`* `unit` *`logical-unit-number`* `family` *`family`* `service]` hierarchy level.

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

RELATED DOCUMENTATION

- Associating Service Sets with Interfaces in a Dynamic Profile | 387
- Verifying and Managing Service Sets Information | 388
- Understanding Service Sets
- Applying Filters and Services to Interfaces

Associating Service Sets with Interfaces in a Dynamic Profile

After you configure a service set, you use a dynamic profile to dynamically associate the service set with interfaces. You reference the filter in the `interfaces` stanza of a dynamic profile. When the dynamic profile instantiates a subscriber session, the router applies the terms of the filter to the interface.

To apply a service set to an interface in a dynamic profile:

1. Access the dynamic profile you want to use.

```
[edit]
user@host# edit dynamic-profiles myProfile
```

2. Specify the interface for the dynamic profile—use the dynamic interface variable.

```
[edit dynamic-profiles myProfile]
user@host# edit interfaces $junos-interface-ifd-name
```

3. Specify the underlying interface—use the unit number variable.

```
[edit dynamic-profiles myProfile interfaces “$junos-interface-ifd-name”]
user@host# edit unit $junos-underlying-interface-unit
```

4. Specify the family. Dynamic service sets are supported only on `family inet` (IPv4).

```
[edit dynamic-profiles myProfile interfaces “$junos-interface-ifd-name” unit “$junos-
underlying-interface-unit”]
user@host# edit family inet
```

5. Specify the input and output service sets that you want to apply to the interface.

```
[edit dynamic-profiles myProfile interfaces “$junos-interface-ifd-name” unit “$junos-
underlying-interface-unit” family inet]
user@host# set service input service-set inputService_200
user@host# set service input post-service-filter postService_15
user@host# set service output service-set outputService_320
```

RELATED DOCUMENTATION

Dynamic Service Sets Overview | 386
Verifying and Managing Service Sets Information | 388
Configuring Service Sets to be Applied to Services Interfaces
Applying Filters and Services to Interfaces

Verifying and Managing Service Sets Information

IN THIS SECTION

- Purpose | 388
- Action | 388

Purpose

View information for service sets:

Action

- To display summary information for service sets:

```
user@host> show services service-sets summary
```

- To display interface-specific information for service sets:

```
user@host>show services service-sets summary interface interface-name
```

RELATED DOCUMENTATION

Dynamic Service Sets Overview | 386
Associating Service Sets with Interfaces in a Dynamic Profile | 387

CHAPTER 25

Configuring Rate-Limiting Premium and Non-Premium Traffic on an Interface Using Hierarchical Policers

IN THIS CHAPTER

- Methods for Regulating Traffic by Applying Hierarchical Policers | 390
- Hierarchical Policer Applied as Filter Action | 393
- Example: Configuring Hierarchical Policers to Limit Rates of Services in a Static Environment | 394

Methods for Regulating Traffic by Applying Hierarchical Policers

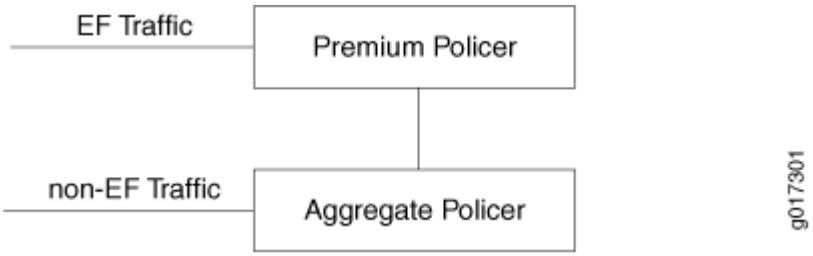
You can deploy policers to enforce service level agreements limiting the input rate at the edge, and at the boundary between domains, to guarantee an equitable deployment of the service among the different domains. Policers determine whether each packet conforms (falls within the traffic contract), exceeds (using up the excess burst capacity), or violates (totally out of the traffic contract rate) the configured traffic policies, and then sets the prescribed action.

Hierarchical policers rate-limit premium traffic separately from the aggregate traffic on an interface as determined by different configured rates. You can use a hierarchical policer to rate-limit ingress Layer 2 traffic at a physical or logical interface and apply different policing actions based on whether the traffic or packets are classified for expedited forwarding (EF) or for a lower priority, such as non-expedited forwarding (non-EF).

Hierarchical policers provide cross-functionality between the configured physical interface and the Packet Forwarding Engine. You can apply a hierarchical policer for premium and aggregate (premium plus normal) traffic levels to a logical interface.

Hierarchical policing uses two token buckets, one for premium (EF) traffic and one for aggregate (non-EF) traffic, as shown in Figure 6 on page 391.

Figure 6: Hierarchical Policer



The class-of-service (CoS) configuration determines which traffic is EF and which is non-EF. Logically, hierarchical policing is achieved by chaining two policers.

- Premium policer—You configure the premium policer with traffic limits for high-priority EF traffic only: a guaranteed bandwidth and a corresponding burst-size limit. EF traffic is categorized as nonconforming when its average arrival rate exceeds the guaranteed bandwidth and its average packet size exceeds the premium burst-size limit. For a premium policer, the only configurable action for nonconforming traffic is to discard the packets.

- Aggregate policer—You configure the aggregate policer (also known as a logical interface policer) with an aggregate bandwidth (to accommodate both high-priority EF traffic up to the guaranteed bandwidth and normal-priority non-EF traffic) and a burst-size limit for non-EF traffic only. Non-EF traffic is categorized as nonconforming when its average arrival rate exceeds the amount of aggregate bandwidth not currently consumed by EF traffic and its average packet size exceeds the burst-size limit defined in the aggregate policer. For an aggregate policer, the configurable actions for nonconforming traffic are to discard the packets, assign a forwarding class, or assign a packet loss priority (PLP) level.

NOTE: You must configure the bandwidth limit of the premium policer at or below the bandwidth limit of the aggregate policer. If the two bandwidth limits are equal, then only non-EF traffic passes through the interface unrestricted; no EF traffic arrives at the interface.

Ingress traffic is first classified into EF and non-EF traffic prior to applying a policer. EF traffic is guaranteed the bandwidth specified as the premium bandwidth limit, while non-EF traffic is rate-limited to the amount of aggregate bandwidth not currently consumed by the EF traffic. Non-EF traffic is rate-limited to the entire aggregate bandwidth only while no EF traffic is present.

Hierarchical policing uses two token buckets, one for aggregate (non-EF) traffic and one for premium (EF) traffic. In Figure 6 on page 391, the premium policer policies EF traffic and the aggregate policer polices non-EF traffic. In the sample configuration that follows, the hierarchical policer is configured with the following components:

- Premium policer has a bandwidth limit set to 2 Mbps, burst-size limit set to 50 KB, and nonconforming action set to discard packets.
- Aggregate policer has a bandwidth limit set to 10 Mbps, burst-size limit set to 100 KB, and nonconforming action set to mark high PLP.

```
[edit]
user@host# show dynamic-profiles firewall
hierarchical-policer policer-agg-prem {
    aggregate {
        if-exceeding {
            bandwidth-limit 10m;
            burst-size-limit 100k;
        }
        then {
            loss-priority high;
        }
    }
    premium {
        if-exceeding {
            bandwidth-limit 2m;
            burst-size-limit 50k;
        }
        then {
            discard;
        }
    }
}
```

EF traffic is guaranteed a bandwidth of 2 Mbps. Bursts of EF traffic—EF traffic that arrives at the interface at rates above 2 Mbps—can also pass through the interface, provided that sufficient tokens are available in the 50 KB burst bucket. When no tokens are available, EF traffic is rate-limited using the discarded action associated with the premium policer.

Non-EF traffic is metered to a bandwidth limit that ranges between 8 Mbps and 10 Mbps, depending on the average arrival rate of the EF traffic. Bursts of non-EF traffic—non-EF traffic that arrives at the interface at rates above the current limit for non-EF traffic—also pass through the interface, provided that sufficient tokens are available in the 100 KB bandwidth bucket. Aggregate traffic in excess of the currently configured bandwidth or burst size are rate-limited using the action specified for the aggregate policer, which in this example is set to a high PLP.

The premium traffic is policed by both the premium policer and aggregate policer. Although the premium policer rate-limits the premium traffic, the aggregate policer decrements the credits but does not drop

the packets. The aggregate policer rate-limits the non-premium traffic. Therefore, the premium traffic is assured to have the bandwidth configured for premium, and the non-premium traffic is policed to the remaining bandwidth.

RELATED DOCUMENTATION

Example: Configuring Hierarchical Policers to Limit Rates of Services in a Static Environment | 394

Hierarchical Policer Applied as Filter Action | 393

Hierarchical Policer Applied as Filter Action

After you define firewall filters and policers, you must apply them to take effect.

- You can apply the same firewall filter to multiple interfaces at the same time. By default on MX Series routers, these filters aggregate their counters and policing actions when those interfaces share a Packet Forwarding Engine. To override this behavior and make each counter or policer function specific to each interface application, include the `interface-specific` statement in the firewall filter.

  ```
  [edit dynamic-profiles profile-name firewall family family filter filter-name
  user@host# set interface-specific
  ```

  Interface-specific filters are particularly useful for IPTV services where television services are delivered using the IP suite over a packet-switched network instead of being delivered through traditional satellite signal and cable television formats.

  NOTE: When you define an interface-specific filter, you must limit the filter name to no more than 52 bytes. Firewall filter names are restricted to 64 bytes in length and interface-specific filters have the specific-name appended to them to differentiate their counters and policing actions. If the automatically generated filter instance name exceeds this maximum length, the system may reject the filter's instance name.

- Alternatively, you can apply a policer to a logical interface either directly or indirectly through a filter that references the policer function. By default, policers are *term-specific*. Junos OS creates a separate policer instance when the same policer is referenced in multiple terms of a firewall filter.

Hierarchical policers provide cross-functionality between the configured physical interface and the Packet Forwarding Engine for provider edge applications. You can apply a hierarchical policer as a filter action for premium and aggregate (premium plus normal) traffic levels to a logical interface. Additionally,

an interface-specific filter can have a hierarchical policer as a filter action whether or not the hierachical policer is a logical interface policer.

A logical interface policer (also known as an aggregate policer) can police the traffic from multiple protocol families without requiring a separate instantiation of a policer for each such family on the logical interface. You define a logical interface policer by including the `logical-interface-policer` statement when defining the policer.

```
[edit dynamic-profiles profile-name firewall policer policer-name
user@host# set logical-interface-policer
```

To apply a logical interface policer on an MX Series router as an action in a firewall filter term, you must specify both the `interface-specific` statement in the firewall filter and the `logical-interface-policer` statement in the related policer. Using a filter to evoke a logical interface filter has the added benefits of increased match flexibility as well as support for two-color policer styles (a policer that classifies traffic into two groups using only the `bandwidth-limit` and `burst-size-limit` parameters), which can only be attached at the family level through a filter action.

NOTE: A non-interface-specific filter can only have a hierarchical policer if no logical interface-specific filter action is specified.

RELATED DOCUMENTATION

Methods for Regulating Traffic by Applying Hierarchical Policers | 390

Example: Configuring Hierarchical Policers to Limit Rates of Services in a Static Environment | 394

Example: Configuring Hierarchical Policers to Limit Rates of Services in a Static Environment

IN THIS SECTION

- Requirements | 395
- Overview | 395
- Configuration | 397

Verification | 407

This example shows how to configure a hierarchical policer and apply the policer to ingress Layer 2 traffic at a logical interface on an MX Series router.

Requirements

Before you begin, be sure that your environment meets the following requirements:

- The interface on which you apply the hierarchical policer is an interface hosted on an MX Series router.
- No other policer is applied to the input of the interface on which you apply the hierarchical policer.
- You are aware that, if you apply the hierarchical policer to logical interface on which an input filter is also applied, the policer is executed first.

Overview

In this example, you configure a hierarchical policer and apply the policer to ingress Layer 2 traffic at a logical interface. Table 35 on page 395 describes the hierarchy levels at which you can configure and apply hierarchical policers on logical and physical interfaces.

Table 35: Hierarchical Policer Configuration and Application Summary

Policer Configuration	Layer 2 Application	Key Points
Hierarchical Policer		
Hierarchically rate-limits Layer 2 ingress traffic for all protocol families. Cannot be applied to egress traffic, Layer 3 traffic, or at a specific protocol level of the interface hierarchy. Supported on interfaces on Dense Port Concentrators (DPCs) in MX Series routers.		

Table 35: Hierarchical Policer Configuration and Application Summary *(Continued)*

Policer Configuration	Layer 2 Application	Key Points

Aggregate and premium policing components of a hierarchical policer:
 	
`[edit dynamic-profiles` *`profile-name`* `firewall]`
`hierarchical-policer` *`policer-name`* `{`
`  aggregate {`
`  if-exceeding {`
`  bandwidth-limit` *`bps`*`;`
`  burst-size-limit` *`bytes`*`;`
`  }`
`  then {`
`  discard;`
`  forwarding-class` *`class-name`*`;`
`  loss-priority` *`supported-value`*`;`
`  }`
`  }`
`  premium {`
`  if-exceeding {`
`  bandwidth-limit` *`bps`*`;`
`  burst-size-limit` *`bytes`*`;`
`  }`
`  then {`
`  discard;`
`  }`
`  }`
`}`
	Option A (physical interface)—Apply directly to Layer 2 input traffic on a physical interface:
	
	`[edit dynamic-profiles` *`profile-name`* `interfaces]`
	`interface-name` `{`
	`  layer2-policer {`
	`  input-hierarchical-policer` *`policer-name`*`;`
	`  }`
	`}`
		Hierarchically rate-limit Layer 2 ingress traffic for all protocol families and logical interfaces configured on a physical interface.
		
		Include the `layer2-policer` configuration statement at the `[edit dynamic-profiles` *`profile-name`* `interfaces` *`interface-name`*`]` hierarchy level.
		
		NOTE: If you apply a hierarchical policer at a physical interface, you cannot also apply a hierarchical policer to any of the member logical interfaces.

	Option B (logical interface)—Apply directly to Layer 2 input traffic on a logical interface:
	
	`[edit dynamic-profiles` *`profile-name`* `interfaces]`
	`interface-name` `{`
	`  unit` *`unit-number`* `{`
	`  layer2-policer {`
	`  input-hierarchical-policer` *`policer-name`*`;`
	`  }`
	`  }`
	`}`
		Hierarchically rate-limit Layer 2 ingress traffic for all protocol families configured on a specific logical interface.
		
		Include the `layer2-policer` configuration statement at the `[edit dynamic-profiles` *`profile-name`* `interfaces` *`interface-name`* `unit` *`unit-number`*`]` hierarchy level.
		
		NOTE: You must configure at least one protocol family for the logical interface.

You apply the policer to the Gigabit Ethernet logical interface ge-1/2/0.0, which you configure for IPv4 traffic. When you apply the hierarchical policer to the logical interface, IPv4 traffic is hierarchically rate-

limited. If you choose to apply the hierarchical policer to physical interface ge-1/2/0, hierarchical policing applies to IPv4 traffic across the logical interface as well.

Configuration

IN THIS SECTION

- CLI Quick Configuration | 397
- Configuring a Basic Dynamic Profile for Subscriber Management | 398
- Configuring the Interfaces | 400
- Configuring the Firewall Filter | 401
- Configuring the Forwarding Classes | 403
- Configuring the Hierarchical Policer | 404
- Applying the Hierarchical Policer to Layer 2 Ingress Traffic at a Physical or Logical Interface | 406

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*.

To configure this example, perform the following tasks:

CLI Quick Configuration

To quickly configure this example, copy the following configuration commands into a text file, remove any line breaks, and then paste the commands into the CLI at the `[edit]` hierarchy level.

```
set dynamic-profiles basic-profile
set dynamic-profiles basic-profile interfaces “$junos-interface-ifd-name”
set dynamic-profiles basic-profile interfaces "$junos-interface-ifd-name" unit “$junos-
underlying-interface-unit”
set dynamic-profiles basic-profile interfaces "$junos-interface-ifd-name" unit $junos-underlying-
interface-unit family inet
set dynamic-profiles interfaces ge-1/2/0 unit 0 family inet address 203.0.113.80/31
set dynamic-profiles basic-profile firewall family inet filter hierarch-filter
set dynamic-profiles basic-profile firewall family inet filter hierarch-filter interface-specific
set dynamic-profiles basic-profile firewall family inet filter hierarch-filter term match-ip1
set dynamic-profiles basic-profile firewall family inet filter hierarch-filter term match-ip2
set dynamic-profiles basic-profile firewall family inet filter hierarch-filter term match-ip1
from precedence critical-ecp protocol
```

```
set dynamic-profiles basic-profile firewall family inet filter hierarch-filter term match-ip1
from protocol tcp
set dynamic-profiles basic-profile firewall family inet filter hierarch-filter term match-ip1
then hierarchical-policer hp1-share filter-specific
set dynamic-profiles basic-profile firewall family inet filter hierarch-filter term match-ip2
from precedence internet-control
set dynamic-profiles basic-profile firewall family inet filter hierarch-filter term match-ip2
from protocol tcp
set dynamic-profiles basic-profile firewall family inet filter hierarch-filter term match-ip2
then hierarchical-policer hp2-share
set class-of-service forwarding-classes class fc0 queue-num 0 priority high policing-priority
premium
set class-of-service forwarding-classes class fc1 queue-num 1 priority low policing-priority
normal
set class-of-service forwarding-classes class fc2 queue-num 2 priority low policing-priority
normal
set class-of-service forwarding-classes class fc3 queue-num 3 priority low policing-priority
normal
set dynamic-profiles basic-profile firewall hierarchical-policer policer-agg-prem aggregate if-
exceeding bandwidth-limit 10m burst-size-limit 100k
set dynamic-profiles basic-profile firewall hierarchical-policer policer-agg-prem aggregate then
forwarding-class fc1
set dynamic-profiles basic-profile firewall hierarchical-policer policer-agg-prem premium if-
exceeding bandwidth-limit 2m burst-size-limit 50k
set dynamic-profiles basic-profile firewall hierarchical-policer policer-agg-prem premium then
discard
set dynamic-profiles basic-profile interfaces ge-1/2/0 unit 0 layer2-policer input-hierarchical-
policer policer-agg-prem
```

Configuring a Basic Dynamic Profile for Subscriber Management

Step-by-Step Procedure

A dynamic profile is a set of characteristics, defined in a type of template, that you can use to provide dynamic subscriber access and services for broadband applications. These services are assigned dynamically to interfaces. A basic profile must contain a profile name and have both an interface variable name (such as **$junos-interface-ifd-name**) included at the `[edit dynamic-profiles` *`profile-name`* `interfaces` hierarchy level and logical interface variable name (such as **$junos-underlying-interface-unit** or **$junos-interface-unit**) at the `[edit dynamic-profiles` *`profile-name`* `interfaces` *`variable-interface-name`* `unit]` hierarchy level.

1. Create the new dynamic profile.

```
[edit]
user@host# set dynamic-profiles basic-profile
```

2. Define the *interface-name* variable statement with the internal **$junos-interface-ifd-name** variable used by the router to match the interface name of the receiving interface.

```
[edit dynamic-profiles basic-profile]
user@host# set interfaces “$junos-interface-ifd-name”
```

3. Define the *variable-interface-name* `unit` statement with the internal variable.

   - When referencing an existing interface, specify the **$junos-underlying-interface-unit** variable used by the router to match the unit value of the receiving interface.
   - When creating dynamic interfaces, specify the **$junos-interface-unit** variable used by the router to generate a unit value for the interface.

```
[edit dynamic-profiles basic-profile interfaces "$junos-interface-ifd-name"]
user@host# set unit $junos-underlying-interface-unit
```

   or

```
[edit dynamic-profiles basic-profile interfaces "$junos-interface-ifd-name"]
user@host# set unit $junos-interface-unit
```

4. Define the family address type (inet for IPv4) for the **$junos-interface-unit** variable.

```
[edit dynamic-profiles basic-profile interfaces "$junos-interface-ifd-name" unit $junos-
underlying-interface-unit]
user@host# set family inet
```

Results

Confirm the configuration of the dynamic profile by entering the `show dynamic-profiles` configuration command. If the command output does not display the intended configuration, repeat the instructions in this procedure to correct the configuration.

```
[edit]
user@host# show dynamic-profiles
dynamic-profiles {
    basic-profile {
        interfaces {
            "$junos-interface-ifd-name" {
                unit "$junos-underlying-interface-unit" {
                    family inet;
                }
            }
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring the Interfaces

Step-by-Step Procedure

Define the physical and logical interfaces for this hierarchical policer example.

1. Configure the physical interface.

```
[edit dynamic-profiles basic-profile]
user@host# set interfaces ge-1/2/0
```

2. Configure the logical interface as unit 0 with its IPv4 (inet) protocol family interface.

```
[edit dynamic-profiles basic-profile interfaces ge-1/2/0]
user@host# set unit 0 family inet address 203.0.113.80/31
```

NOTE: If you apply a Layer 2 policer to this logical interface, you must configure at least one protocol family.

Results

Confirm the configuration by entering the `show dynamic-profiles basic-profile interfaces` configuration command. If the command output does not display the intended configuration, repeat the instructions in this procedure to correct the configuration.

```
[edit]
user@host# show dynamic-profiles basic-profile interfaces
ge-1/2/0 {
    unit 0 {
        family inet {
            address 203.0.113.80/31;
        }
    }
}
```

Configuring the Firewall Filter

Step-by-Step Procedure

To configure a hierarchical policer as a filter action, you must first configure a firewall filter.

1. Configure the family address type (inet for IPv4) for the firewall filter and specify the filter name.

   We recommend that you name the filter something that indicates the filter's purpose.

   ```
   [edit dynamic-profiles basic-profile]
   user@host# set firewall family inet filter hierarch-filter
   ```

2. To override the aggregation of the counters and policing actions and make each counter or policy function specific to each interface application, include the `interface-specific` statement in the filter.

   ```
   [edit dynamic-profiles basic-profile firewall family inet filter hierarch-filter]
   user@host# set interface-specific
   ```

3. Specify the term names for the filter.

   Make each term name unique and represent what its function is.

```
[edit dynamic-profiles basic-profile firewall family inet filter hierarch-filter]
user@host# set term match-ip1
user@host# set term match-ip2
```

4. In each firewall filter term, specify the conditions used to match components of a packet.

   Configure the first term to match IPv4 packets received through TCP with the IP precedence field critical-ecp (0xa0) protocol, and apply the hierarchical policer as a filter action.

```
[edit dynamic-profiles basic-profile firewall family inet filter hierarch-filter term match-
ip1]
user@host# set from precedence critical-ecp protocol
user@host# set from protocol tcp
```

5. Specify the actions to take when the packet matches all of the conditions in the first term. Enable all hierarchical policers in one filter to share the same policer instance in the Packet Forward Engine.

```
[edit dynamic-profiles basic-profile firewall family inet filter hierarch-filter term match-
ip1]
user@host# set then hierarchical-policer hp1-share filter-specific
```

6. Configure the second term to match IPv4 packets received through TCP with the IP precedence field internet-control (0xc0), and apply the hierarchical policer as a filter action.

```
[edit dynamic-profiles basic-profile firewall family inet filter hierarch-filter term match-
ip2]
user@host# set from precedence internet-control
user@host# set from protocol tcp
```

7. Specify the actions to take when the packet matches all of the conditions in the second term.

```
[edit dynamic-profiles basic-profile firewall family inet filter inet-filter term match-ip2]
user@host# set then hierarchical-policer hp2-share
```

Results

Confirm the configuration by entering the `show dynamic-profiles basic-profile firewall` configuration command. If the command output does not display the intended configuration, repeat the instructions in this procedure to correct the configuration.

```
[edit]
user@host# show dynamic-profiles basic-profile firewall
family inet {
    filter hierarch-filter {
        interface-specific;
        term match-ip1 {
            from {
                precedence critical-ecp protocol;
                protocol tcp;
            }
            then hierarchical-policer hp1-share;
        }
        term match-ip2 {
            from {
                precedence internet-control;
                protocol tcp;
            }
            then hierarchical-policer hp2-share;
        }
    }
}
```

Configuring the Forwarding Classes

Step-by-Step Procedure

Define forwarding classes referenced as aggregate policer actions. For hierarchical policers to work, ingress traffic must be correctly classified into premium and non-premium buckets. Some class-of-service (CoS) configuration is required because the hierarchical policer must be able to separate premium/expedited forwarding (EF) traffic from non-premium/non-EF traffic.

1. Enable configuration of the forwarding classes.

```
[edit]
user@host# set class-of-service forwarding-classes
```

2. Define CoS forwarding classes to include the designation of which forwarding class is premium. This defaults to the forwarding class associated with EF traffic.

```
[edit class-of-service forwarding-classes]
user@host# set class fc0 queue-num 0 priority high policing-priority premium
user@host# set class fc1 queue-num 1 priority low policing-priority normal
user@host# set class fc2 queue-num 2 priority low policing-priority normal
user@host# set class fc3 queue-num 3 priority low policing-priority normal
```

Results

Confirm the configuration of the forwarding classes referenced as aggregate policer actions by entering the `show class-of-service` configuration command. If the command output does not display the intended configuration, repeat the instructions in this procedure to correct the configuration.

```
[edit]
user@host# show class-of-service
forwarding-classes {
    class fc0 queue-num 0 priority high policing-priority premium;
    class fc1 queue-num 1 priority low policing-priority normal;
    class fc2 queue-num 2 priority low policing-priority normal;
    class fc3 queue-num 3 priority low policing-priority normal;
}
```

Configuring the Hierarchical Policer

Step-by-Step Procedure

Configure the aggregate and premium policing components of a hierarchical policer.

1. Enable configuration of the hierarchical policer.

```
[edit dynamic-profiles basic-profile]
user@host# set firewall hierarchical-policer policer-agg-prem
```

2. Configure the aggregate policer to have a bandwidth limit set to 10 Mbps, burst-size limit set to 100 KB, and nonconforming action set to change the forwarding class to fc1.

```
[edit dynamic-profiles basic-profile firewall hierarchical-policer policer-agg-prem]
user@host# set aggregate if-exceeding bandwidth-limit 10m burst-size-limit 100k
user@host# set aggregate then forwarding-class fc1
```

NOTE: For aggregate policers, the configurable actions for a packet in a nonconforming flow are to discard the packet, change the loss priority, or change the forwarding class.

3. Configure the premium policer to have a bandwidth limit set to 2 Mbps, burst-size limit set to 50 KB, and nonconforming action set to discard packets.

```
[edit dynamic-profiles basic-profile firewall hierarchical-policer policer-agg-prem]
user@host# set premium if-exceeding bandwidth-limit 2m burst-size-limit 50k
user@host# set premium then discard
```

NOTE: The bandwidth limit for the premium policer must not be greater than that of the aggregate policer. For the premium policers, the only configurable action for a packet in a nonconforming traffic flow is to discard the packet.

Results

Confirm the configuration of the hierarchical policer by entering the `show dynamic-profiles basic-profile firewall` configuration command. If the command output does not display the intended configuration, repeat the instructions in this procedure to correct the configuration.

```
[edit]
user@host# show dynamic-profiles basic-profile firewall
hierarchical-policer policer-agg-prem {
    aggregate {
```

```
        if-exceeding {
            bandwidth-limit 10m;
            burst-size-limit 100k;
        }
        then {
            forwarding-class fc1;
        }
    }
    premium {
        if-exceeding {
            bandwidth-limit 2m;
            burst-size-limit 50k;
        }
        then {
            discard;
        }
    }
}
```

Applying the Hierarchical Policer to Layer 2 Ingress Traffic at a Physical or Logical Interface

Step-by-Step Procedure

You can apply policers directly to an interface or applied through a filter to affect only matching traffic. In most cases, you can invoke a policing function at ingress, egress, or in both directions.

- For physical interfaces, a hierarchical policer uses a single policer instance to rate-limit all logical interfaces and protocol families configured on a physical interface, even if the logical interfaces have mutually exclusive families such as inet or bridge.
- For logical interfaces, a hierarchical policer can police the traffic from multiple protocol families without requiring a separate instantiation of a policer for each such family on the logical interface.

To hierarchically rate-limit Layer 2 ingress traffic for IPv4 traffic on logical interface ge-1/2/0.0, reference the policer from the logical interface configuration.

1. Configure the logical interface.

```
[edit dynamic-profiles basic-profile]
user@host# set interfaces ge-1/2/0 unit 0
```

When you apply a policer to Layer 2 traffic at a logical interface, you must define at least one protocol family for the logical interface.

2. Apply the policer to the logical interface.

```
[edit dynamic-profiles basic-profile interfaces ge-1/2/0 unit 0]
user@host# set layer2-policer input-hierarchical-policer policer-agg-prem
```

Alternatively, to hierarchically rate-limit Layer 2 ingress traffic for all protocol families and for *all logical interfaces* configured on physical interface ge-1/2/0, reference the policer from the physical interface configuration.

Results

Confirm the configuration of the hierarchical policer by entering the `show dynamic-profiles basic-profile interfaces` configuration command. If the command output does not display the intended configuration, repeat the instructions in this procedure to correct the configuration.

```
[edit]
user@host# show dynamic-profiles basic-profile interfaces
ge-1/2/0 {
    unit 0 {
        layer2-policer {
            input-hierarchical-policer policer-agg-prem;
        }
        family inet {
            address 203.0.113.80/31;
        }
    }
}
```

Verification

IN THIS SECTION

- Displaying Traffic Statistics for the Interface | 408
- Displaying Number of Packets Policed by the Specified Policer | 410

Confirm that the configuration is working properly.

Displaying Traffic Statistics for the Interface

Purpose

Verify the traffic flow through the physical interface.

Action

Use the `show interfaces` operational mode command for physical interface ge-1/2/0, and include the `detail` or `extensive` option.

```
user@host> show interfaces ge-1/2/0 extensive

Physical interface: ge-1/2/0, Enabled, Physical link is Down
  Interface index: 156, SNMP ifIndex: 630, Generation: 159
  Link-level type: Ethernet, MTU: 1514, MRU: 1522, Speed: 1000mbps, BPDU Error: None, MAC-
REWRITE Error: None, Loopback: Disabled,
  Source filtering: Disabled, Flow control: Enabled, Auto-negotiation: Enabled, Remote fault:
Online
  Pad to minimum frame size: Disabled
  Device flags   : Present Running Down
  Interface flags: Hardware-Down SNMP-Traps Internal: 0x4000
  Link flags     : None
  CoS queues     : 8 supported, 8 maximum usable queues
  Schedulers     : 0
  Hold-times     : Up 0 ms, Down 0 ms
  Current address: 00:00:5E:00:53:4c, Hardware address: 00:00:5E:00:53:4c
  Last flapped   : 2014-11-10 13:36:25 EST (01:26:30 ago)
  Statistics last cleared: Never
  Traffic statistics:
   Input  bytes  :                   0                   0 bps
   Output bytes  :                  42                   0 bps
   Input  packets:                   0                   0 pps
   Output packets:                   1                   0 pps
   IPv6 transit statistics:
    Input  bytes  :                  0
    Output bytes  :                  0
    Input  packets:                  0
    Output packets:                  0
  Dropped traffic statistics due to STP State:
```

```
   Input  bytes  :                    0
   Output bytes  :                    0
   Input  packets:                    0
   Output packets:                    0
  Input errors:
    Errors: 0, Drops: 0, Framing errors: 0, Runts: 0, Policed discards: 0, L3 incompletes: 0, L2
channel errors: 0, L2 mismatch timeouts: 0,
    FIFO errors: 0, Resource errors: 0
  Output errors:
    Carrier transitions: 0, Errors: 0, Drops: 0, Collisions: 0, Aged packets: 0, FIFO errors: 0,
HS link CRC errors: 0, MTU errors: 0,
    Resource errors: 0
  Egress queues: 8 supported, 8 in use
  Queue counters:       Queued packets  Transmitted packets      Dropped packets
    0                              0                    0                    0
    1                              0                    0                    0
    2                              0                    0                    0
    3                              0                    0                    0
    4                              0                    0                    0
    5                              0                    0                    0
    6                              0                    0                    0
    7                              0                    0                    0
  Queue number:         Mapped forwarding classes
    0                   best-effort
    1                   expedited-forwarding
    2                   assured-forwarding
    3                   network-control
    4                   be1
    5                   ef1
    6                   af1
    7                   nc1
  Active alarms  : LINK
  Active defects : LINK
  MAC statistics:                      Receive         Transmit
    Total octets                             0                0
    Total packets                            0                0
    Unicast packets                          0                0
    Broadcast packets                        0                0
    Multicast packets                        0                0
    CRC/Align errors                         0                0
    FIFO errors                              0                0
    MAC control frames                       0                0
    MAC pause frames                         0                0
```

```
    Oversized frames                          0
    Jabber frames                             0
    Fragment frames                           0
    VLAN tagged frames                        0
    Code violations                           0
    Total errors                              0                0
  Filter statistics:
    Input packet count                        0
    Input packet rejects                      0
    Input DA rejects                          0
    Input SA rejects                          0
    Output packet count                                        0
    Output packet pad count                                    0
    Output packet error count                                  0
    CAM destination filters: 0, CAM source filters: 0
  Autonegotiation information:
    Negotiation status: Incomplete
  Packet Forwarding Engine configuration:
    Destination slot: 0 (0x00)
  CoS information:
    Direction : Output
    CoS transmit queue               Bandwidth              Buffer Priority   Limit
                               %            bps     %           usec
    0 best-effort             95      950000000    95              0      low    none
    3 network-control          5       50000000     5              0      low    none
  Interface transmit statistics: Disabled
```

Meaning

The command output section for **Traffic statistics** lists the number of bytes and packets received and transmitted on the interface.

Displaying Number of Packets Policed by the Specified Policer

Purpose

Verify the number of packets evaluated by the policer. Premium policer counters are not supported.

Action

Use the `show policer` operational mode command and optionally specify the name of the policer `policer-agg-prem`. The command output displays the number of packets evaluated by the specified policer in each direction.

```
user@host> show policer policer-agg-prem
Policers:
Name                                                Bytes              Packets
policer-agg-prem-ge-1/2/0.0-inet-i                10372300               103723
```

The **-inet-i** suffix denotes a policer applied to IPv4 input traffic. In this example, the policer is applied to input traffic only.

Meaning

The command output displays the number of packets evaluated by the specified policer in each direction.

RELATED DOCUMENTATION

Methods for Regulating Traffic by Applying Hierarchical Policers | 390

Hierarchical Policer Applied as Filter Action | 393

CHAPTER 26

Configure Rate-Limiting Traffic on an Interface Using Enhanced Hierarchical Policers

IN THIS CHAPTER

- Enhanced Hierarchical Policer Overview (Junos OS Evolved) | 412
- Configure Enhanced Hierarchical Policer (Junos OS Evolved) | 414
- Verify Enhanced Hierarchical Policer Configuration | 423

Enhanced Hierarchical Policer Overview (Junos OS Evolved)

SUMMARY

Enhanced hierarchical policer configuration enables traffic rate limiting of data packets based on the traffic priority. You can configure enhanced hierarchical policer under dynamic profile to rate limit traffic for subscribers. Traffic policing is supported at four levels of hierarchies with respect to the traffic priority- High, Medium-High, Medium-Low, and Low.

IN THIS SECTION

- Guidelines for Configuring Enhanced Hierarchical Policer | 413
- **Enhanced Hierarchical Policer Applied as Filter Action | 414**

In an enhanced hierarchical policer configuration, you can define up to four policers. Each policer maps to a traffic priority. The four traffic priorities, arranged as per their order of precedence are High, Medium-High, Medium-Low, and Low. The traffic priorities are hierarchical – High is the traffic priority with the highest precedence and Low is the traffic priority with the lowest precedence. It implies that a policer defined for the High traffic priority has a higher precedence than the rest of the policers or a policer defined for the Low traffic priority has a lower precedence than the rest of the policers.

Each policer is allotted a Confirmed Information Rate (CIR) and Maximum Confirmed Information Rate (MAX CIR). Residue bandwidth or unused bandwidth is carried over to lower precedence policers. As the Table 36 on page 413 shows, medium-high policer inherits unused bandwidth from high policer. Medium-low policer inherits from high and medium-high policers. Low policer inherits from the other

three higher precedence policers. It is recommended that MAX CIR of a particular level is equal to the CIR of current level + combined CIR of previous/top levels.

Table 36: Example enhanced hierarchical policer configuration

Policer Configurations		
Policer-level/traffic-priority	**CIR**	**MAX CIR**
high	10mbps	10mbps
medium-high	10mbps	20mbps
medium-low	10mbps	30mbps
low	10mbps	40mbps

Guidelines for Configuring Enhanced Hierarchical Policer

- The CIR and Max CIR values must be the same for the policer with the highest precedence.
- The minimum bandwidth rate that you can allot for CIR or MAX CIR is 22KB.
- It is mandatory to configure all the levels (four) of an enhanced hierarchical policer with respective policer bandwidth rates and burst size. However, the firewall filter can refer to any number (one to four) of policers based on the requirement. If you do not want all the four levels of enhanced hierarchical policer but need only two levels, then all the firewall filters can refer to only two active levels. If you do not want all the four levels of enhanced hierarchical policer, but need only one level, then all the firewall filters can refer to only one level. If there is no requirement to configure all four levels, the unwanted levels must specify least supported CIR, MAX CIR and CBS rates. It is recommended that firewall filter terms not be mapped to these unwanted levels.
- Configuration of both hierarchical policer and enhanced hierarchical policer inside Dynamic Profile is not allowed.
- You must configure each enhanced hierarchical policer level with the action `discard` to drop the packets exceeding the configured bandwidth.

Enhanced Hierarchical Policer Applied as Filter Action

You can apply enhanced hierarchical policer to a logical interface as a filter action for aggregate traffic levels. Junos OS Evolved supports the use of logical-interface-policer (also known as an aggregate policer) within the enhanced hierarchical policer configuration. It enables traffic rate limiting at the interface level across multiple protocol families (`inet` and `inet6`), without requiring separate policer instances for each family.

NOTE:

- You can configure logical-interface-policer only for families inet and inet6.
- On Junos OS Evolved devices, logical-interface-policer applies only for input direction.

To apply the policer function specific to an interface, include the `interface-specific` statement in the firewall filter configuration.

```
[edit dynamic-profiles profile-name firewall family family filter filter-name]
user@host# set interface-specific
```

You can define a logical interface policer by including the `logical-interface-policer` statement when configuring the enhanced hierarchical policer.

```
[edit dynamic-profiles profile-name firewall enhanced-hierarchical-policer policer-name]
user@host# set logical-interface-policer
```

Configure Enhanced Hierarchical Policer (Junos OS Evolved)

SUMMARY

This example shows how to configure enhanced-hierarchical-policer under the dynamic profile using four levels of traffic priority hierarchies– high, medium-high, medium-low, and low. To know more about enhanced hierarchical policer functionality, see Enhanced Hierarchical Policer Overview (Junos OS Evolved).

The following steps require you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode.

1. Configure prefix list to enable the dynamic profile to classify traffic based on IP prefixes. This allows the enhanced hierarchical policer to selectively apply rate-limiting policies to those matched prefixes.

```
[edit]
set policy-options prefix-list v4-16-17 172.16.100.100/32
set policy-options prefix-list v4-16-17 172.17.100.100/32
set policy-options prefix-list v4-18-19 172.18.100.100/32
set policy-options prefix-list v4-18-19 172.19.100.100/32
set policy-options prefix-list v4-20-21 172.20.100.100/32
set policy-options prefix-list v4-20-21 172.21.100.100/32
set policy-options prefix-list v4-22-23 172.22.100.100/32
set policy-options prefix-list v4-22-23 172.23.100.100/32
set policy-options prefix-list v6-16-17 172:16:100:100::0/64
set policy-options prefix-list v6-16-17 172:17:100:100::0/64
set policy-options prefix-list v6-18-19 172:18:100:100::0/64
set policy-options prefix-list v6-18-19 172:19:100:100::0/64
set policy-options prefix-list v6-20-21 172:20:100:100::0/64
set policy-options prefix-list v6-20-21 172:21:100:100::0/64
set policy-options prefix-list v6-22-23 172:22:100:100::0/64
set policy-options prefix-list v6-22-23 172:23:100:100::0/64
```

2. Create a dynamic profile.

```
[edit]
set dynamic-profiles pppoe-client-enh-hpol-profile-scaling
```

3. Configure variables within the dynamic profile. They act as placeholders or keys with dynamically assigned values sourced from a RADIUS server during subscriber authentication.

```
[edit dynamic-profiles pppoe-client-enh-hpol-profile-scaling]
set variables upstream-inet uid
set variables mef_prof uid
set variables upstream-inet6 uid
```

4. Configure the default values for the predefined variables. These default values are used when RADIUS server does not supply a value.

```
[edit dynamic-profiles pppoe-client-enh-hpol-profile-scaling]
set predefined-variable-defaults hpolicer-high-cir 50k
set predefined-variable-defaults hpolicer-high-max-cir 50k
set predefined-variable-defaults hpolicer-high-committed-burst 1k
set predefined-variable-defaults hpolicer-med-high-cir 50k
set predefined-variable-defaults hpolicer-med-high-max-cir 65k
set predefined-variable-defaults hpolicer-med-high-committed-burst 1k
set predefined-variable-defaults hpolicer-med-low-cir 50k
set predefined-variable-defaults hpolicer-med-low-max-cir 80k
set predefined-variable-defaults hpolicer-med-low-committed-burst 1k
set predefined-variable-defaults hpolicer-low-cir 50k
set predefined-variable-defaults hpolicer-low-max-cir 100k
set predefined-variable-defaults hpolicer-low-committed-burst 1k
```

5. Create enhanced hierarchical policer under the dynamic profile hierarchy. Also, include `logical-interface-policer` statement to define traffic rate limiting of input traffic for both inet and inet6 families on the same logical interface without needing to create multiple instances of the policer.

```
[edit dynamic-profiles pppoe-client-enh-hpol-profile-scaling]
set firewall enhanced-hierarchical-policer "$mef_prof" logical-interface-policer
```

6. Define committed information rate (CIR), maximum committed information rate (MAX CIR), and committed burst size (CBS) for the four traffic priorities- high, medium-high, medium-low, and low.

```
[edit dynamic-profiles pppoe-client-enh-hpol-profile-scaling]
set firewall enhanced-hierarchical-policer "$mef_prof" high committed-burst-size "$junos-
hpolicer-high-committed-burst"
set firewall enhanced-hierarchical-policer "$mef_prof" high committed-information-rate
"$junos-hpolicer-high-cir"
set firewall enhanced-hierarchical-policer "$mef_prof" high max-committed-information-rate
"$junos-hpolicer-high-max-cir"
set firewall enhanced-hierarchical-policer "$mef_prof" high then discard
set firewall enhanced-hierarchical-policer "$mef_prof" medium-high committed-burst-size
"$junos-hpolicer-med-high-committed-burst"
set firewall enhanced-hierarchical-policer "$mef_prof" medium-high committed-information-rate
"$junos-hpolicer-med-high-cir"
set firewall enhanced-hierarchical-policer "$mef_prof" medium-high max-committed-information-
rate "$junos-hpolicer-med-high-max-cir"
```

```
set firewall enhanced-hierarchical-policer "$mef_prof" medium-high then discard
set firewall enhanced-hierarchical-policer "$mef_prof" medium-low committed-burst-size
"$junos-hpolicer-med-low-committed-burst"
set firewall enhanced-hierarchical-policer "$mef_prof" medium-low committed-information-rate
"$junos-hpolicer-med-low-cir"
set firewall enhanced-hierarchical-policer "$mef_prof" medium-low max-committed-information-
rate "$junos-hpolicer-med-low-max-cir"
set firewall enhanced-hierarchical-policer "$mef_prof" medium-low then discard
set firewall enhanced-hierarchical-policer "$mef_prof" low committed-burst-size "$junos-
hpolicer-low-committed-burst"
set firewall enhanced-hierarchical-policer "$mef_prof" low committed-information-rate "$junos-
hpolicer-low-cir"
set firewall enhanced-hierarchical-policer "$mef_prof" low max-committed-information-rate
"$junos-hpolicer-low-max-cir"
set firewall enhanced-hierarchical-policer "$mef_prof" low then discard
```

7. Configure firewall filter under dynamic profiles with enhanced hierarchical policer and other required filter actions.

   a. Specify the family address type (inet for IPv4 and/or inet6 for IPV6) for the firewall filter, specify the filter name using a defined variable, and include the `interface-specific` statement to apply the policer function specific to an interface.

   ```
   [edit dynamic-profiles pppoe-client-enh-hpol-profile-scaling]
   set firewall family inet filter "$upstream-inet" interface-specific
   set firewall family inet6 filter "$upstream-inet6" interface-specific
   ```

   b. (Optional) To enable the Packet Forwarding Engine (PFE) to report policer statistics to the RADIUS server, include the `service-accounting` statement. The statistics include the timestamp when the data was collected, along with the total bytes and packets allowed by the policer at each level. The system provides statistics to the RADIUS server for all four policer levels.

   ```
   [edit dynamic-profiles pppoe-client-enh-hpol-profile-scaling]
   set firewall family inet filter "$upstream-inet" service-accounting
   set firewall family inet6 filter "$upstream-inet6" service-accounting
   ```

   c. Specify firewall filter terms based on destination prefixes and enhanced hierarchical policer filter actions.

   ```
   [edit dynamic-profiles pppoe-client-enh-hpol-profile-scaling]
   set firewall family inet filter "$upstream-inet" term platinum from destination-prefix-
   ```

```
list v4-16-17
set firewall family inet filter "$upstream-inet" term platinum then enhanced-hierarchical-
policer "$mef_prof" traffic-priority high
set firewall family inet filter "$upstream-inet" term gold from destination-prefix-list
v4-18-19
set firewall family inet filter "$upstream-inet" term gold then enhanced-hierarchical-
policer "$mef_prof" traffic-priority medium-high
set firewall family inet filter "$upstream-inet" term silver from destination-prefix-list
v4-20-21
set firewall family inet filter "$upstream-inet" term silver then enhanced-hierarchical-
policer "$mef_prof" traffic-priority medium-low
set firewall family inet filter "$upstream-inet" term dflt then enhanced-hierarchical-
policer "$mef_prof" traffic-priority low
set firewall family inet6 filter "$upstream-inet6" term platinum from destination-prefix-
list v6-16-17
set firewall family inet6 filter "$upstream-inet6" term platinum then enhanced-
hierarchical-policer "$mef_prof" traffic-priority high
set firewall family inet6 filter "$upstream-inet6" term gold from destination-prefix-list
v6-18-19
set firewall family inet6 filter "$upstream-inet6" term gold then enhanced-hierarchical-
policer "$mef_prof" traffic-priority medium-high
set firewall family inet6 filter "$upstream-inet6" term silver from destination-prefix-
list v6-20-21
set firewall family inet6 filter "$upstream-inet6" term silver then enhanced-hierarchical-
policer "$mef_prof" traffic-priority medium-low
set firewall family inet6 filter "$upstream-inet6" term dflt then enhanced-hierarchical-
policer "$mef_prof" traffic-priority low
```

8. Configure the logical interface to apply firewall filters.

```
[edit dynamic-profiles pppoe-client-enh-hpol-profile-scaling]
set interfaces pp0 unit "$junos-interface-unit" actual-transit-statistics
set interfaces pp0 unit "$junos-interface-unit" no-traps
set interfaces pp0 unit "$junos-interface-unit" ppp-options chap
set interfaces pp0 unit "$junos-interface-unit" ppp-options pap
set interfaces pp0 unit "$junos-interface-unit" pppoe-options underlying-interface "$junos-
underlying-interface"
set interfaces pp0 unit "$junos-interface-unit" pppoe-options server
set interfaces pp0 unit "$junos-interface-unit" keepalives interval 30
set interfaces pp0 unit "$junos-interface-unit" family inet filter input "$upstream-inet"
set interfaces pp0 unit "$junos-interface-unit" family inet unnumbered-address lo0.0
```

```
set interfaces pp0 unit "$junos-interface-unit" family inet6 filter input "$upstream-inet6"
set interfaces pp0 unit "$junos-interface-unit" family inet6 unnumbered-address lo0.0
```

9. (Optional) Confirm the configuration by entering the `show dynamic-profiles pppoe-client-enh-hpol-profile-scaling` command. If the command output does not display the intended configuration, repeat the instructions in this procedure to correct the configuration.

```
show dynamic-profiles pppoe-client-enh-hpol-profile-scaling | no-more
variables {
    upstream-inet uid;
    mef_prof uid;
    upstream-inet6 uid;
}
predefined-variable-defaults {
    hpolicer-high-cir 50k;
    hpolicer-high-max-cir 50k;
    hpolicer-high-committed-burst 1k;
    hpolicer-med-high-cir 50k;
    hpolicer-med-high-max-cir 65k;
    hpolicer-med-high-committed-burst 1k;
    hpolicer-med-low-cir 50k;
    hpolicer-med-low-max-cir 80k;
    hpolicer-med-low-committed-burst 1k;
    hpolicer-low-cir 50k;
    hpolicer-low-max-cir 100k;
    hpolicer-low-committed-burst 1k;
}
interfaces {
    pp0 {
        unit "$junos-interface-unit" {
            actual-transit-statistics;
            no-traps;
            ppp-options {
                chap;
                pap;
            }
            pppoe-options {
                underlying-interface "$junos-underlying-interface";
                server;
            }
            keepalives interval 30;
            family inet {
```

```
                filter {
                    input "$upstream-inet";
                }
                unnumbered-address lo0.0;
            }
            family inet6 {
                filter {
                    input "$upstream-inet6";
                }
                unnumbered-address lo0.0;
            }
        }
    }
}
firewall {
    family inet {
        filter "$upstream-inet" {
            interface-specific;
            service-accounting;
            term platinum {
                from {
                    destination-prefix-list {
                        v4-16-17;
                    }
                }
                then {
                    enhanced-hierarchical-policer {
                        "$mef_prof";
                        traffic-priority high;
                    }
                }
            }
            term gold {
                from {
                    destination-prefix-list {
                        v4-18-19;
                    }
                }
                then {
                    enhanced-hierarchical-policer {
                        "$mef_prof";
                        traffic-priority medium-high;
                    }
```

```
                }
            }
            term silver {
                from {
                    destination-prefix-list {
                        v4-20-21;
                    }
                }
                then {
                    enhanced-hierarchical-policer {
                        "$mef_prof";
                        traffic-priority medium-low;
                    }
                }
            }
            term dflt {
                then {
                    enhanced-hierarchical-policer {
                        "$mef_prof";
                        traffic-priority low;
                    }
                }
            }
        }
    }
    family inet6 {
        filter "$upstream-inet6" {
            interface-specific;
            service-accounting;
            term platinum {
                from {
                    destination-prefix-list {
                        v6-16-17;
                    }
                }
                then {
                    enhanced-hierarchical-policer {
                        "$mef_prof";
                        traffic-priority high;
                    }
                }
            }
            term gold {
```

```
                from {
                    destination-prefix-list {
                        v6-18-19;
                    }
                }
                then {
                    enhanced-hierarchical-policer {
                        "$mef_prof";
                        traffic-priority medium-high;
                    }
                }
            }
            term silver {
                from {
                    destination-prefix-list {
                        v6-20-21;
                    }
                }
                then {
                    enhanced-hierarchical-policer {
                        "$mef_prof";
                        traffic-priority medium-low;
                    }
                }
            }
            term dflt {
                then {
                    enhanced-hierarchical-policer {
                        "$mef_prof";
                        traffic-priority low;
                    }
                }
            }
        }
    }
    enhanced-hierarchical-policer "$mef_prof" {
        logical-interface-policer;
        high {
            committed-burst-size "$junos-hpolicer-high-committed-burst";
            committed-information-rate "$junos-hpolicer-high-cir";
            max-committed-information-rate "$junos-hpolicer-high-max-cir";
            then discard;
        }
```

```
        medium-high {
            committed-burst-size "$junos-hpolicer-med-high-committed-burst";
            committed-information-rate "$junos-hpolicer-med-high-cir";
            max-committed-information-rate "$junos-hpolicer-med-high-max-cir";
            then discard;
        }
        medium-low {
            committed-burst-size "$junos-hpolicer-med-low-committed-burst";
            committed-information-rate "$junos-hpolicer-med-low-cir";
            max-committed-information-rate "$junos-hpolicer-med-low-max-cir";
            then discard;
        }
        low {
            committed-burst-size "$junos-hpolicer-low-committed-burst";
            committed-information-rate "$junos-hpolicer-low-cir";
            max-committed-information-rate "$junos-hpolicer-low-max-cir";
            then discard;
        }
    }
}
```

Verify Enhanced Hierarchical Policer Configuration

IN THIS SECTION

- Purpose | 423
- Action | 424
- Meaning | 425

Purpose

To verify the configuration and operational status of firewall filters, policers, and counters applied to dynamic subscriber sessions. This helps confirm that the enhanced hierarchical policer and firewall filters are correctly applied and functioning on subscriber interfaces and sessions.

Action

Use the `show subscribers firewall` command to view the results.

```
show subscribers firewall

Filter: __junos_adf_23-pp0.3221225490-inet-in

Filter: __junos_adf_23-pp0.3221225490-inet6-in

Filter: v4-DT-DP_UID1094-pp0.3221225490-in
Counters:
Name
Bytes              Packets
_implct_hpolicer-in_UID1092-High-pp0.3221225490-in
303237200               216598
 implct_hpolicer-in_UID1092-Medium-High-pp0.3221225490-in
0                      0
 implct_hpolicer-in_UID1092-Medium-Low-pp0.3221225490-in
0                      0
_implct_hpolicer-in_UID1092-Low-pp0.3221225490-in
227685335              162636
Enhanced Hierarchical Policers:
Name
Bytes              Packets
    High
123935000               88525
    Medium-High
0                     0
    Medium-Low
0                     0
    Low
47866000                34190

Filter: v6-DT-DP_UID1098-pp0.3221225490-in
Counters:
Name
Bytes              Packets
_implct_hpolicer-in_UID1092-High-pp0.3221225490-in
303237200               216598
 implct_hpolicer-in_UID1092-Medium-High-pp0.3221225490-in
0                      0
```

```
 implct_hpolicer-in_UID1092-Medium-Low-pp0.3221225490-in
0                      0
_implct_hpolicer-in_UID1092-Low-pp0.3221225490-in
227685335               162636
Enhanced Hierarchical Policers:
Name
Bytes              Packets
    High
123935000                88525
    Medium-High
0                      0
    Medium-Low
0                      0
    Low
47866000                34190
```

Meaning

The output indicates which firewall filters and policers are active on subscriber sessions, the amount of traffic matched or dropped by these filters, and the counters associated with each filter term.

If you do not configure a counter for the enhanced hierarchical policer, the system displays implicit counter statistics as shown in the output above.

CHAPTER 27

Monitoring and Managing Firewalls for Subscriber Access

IN THIS CHAPTER

- Verifying and Managing Firewall Filter Configuration | 426
- Enhanced Policer Statistics Overview | 427

Verifying and Managing Firewall Filter Configuration

IN THIS SECTION

- Purpose | 426
- Action | 427

Purpose

View or manage information for firewall filters:

NOTE: The router creates unique names for fast update filters and for filter terms and counters. See *Naming Fast Update Filters* in "Fast Update Filters Overview" on page 344 for information.

Action

- To display statistics for firewall filters:

```
user@host> show firewall
```

- To display firewall log information:

```
user@host> show firewall log
```

- To clear filter counters:

```
user@host> clear firewall all
```

RELATED DOCUMENTATION

Classic Filters Overview | 239

Fast Update Filters Overview | 344

CLI Explorer

Enhanced Policer Statistics Overview

You can use the enhanced policer statistics to analyze traffic for debugging purposes on MPC/MIC interfaces on MX Series routers and Multi-Rate Ethernet Enhanced Queuing IP Services DPC with SFP and XFP.

Enhanced policer statistics provide the following:

- Offered packet statistics for traffic subjected to policing.
- OOS packet statistics for packets that are marked out-of-specification by the policer. Changes to all packets that have out-of-specification actions, such as discard, color marking, or forwarding-class, are included in this counter.
- Transmitted packet statistics for traffic that is not discarded by the policer. When the policer action is discard, the statistics are the same as the within-specification statistics; when the policer action is non-discard (loss-priority or forwarding-class), the statistics are included in this counter.

RELATED DOCUMENTATION

- show policer
- *show firewall*
- *enhanced-policer*

4

PART

Configuring Dynamic Multicast

- Configuring Dynamic IGMP to Support IP Multicasting for Subscribers | **430**
- Configuring Dynamic MLD to Enable Subscribers to Access Multicast Networks | **439**

CHAPTER 28

Configuring Dynamic IGMP to Support IP Multicasting for Subscribers

IN THIS CHAPTER

- Dynamic IGMP Configuration Overview | 430
- Subscriber Management IGMP Model Overview | 431
- Configuring Dynamic DHCP Client Access to a Multicast Network | 432
- Example: IGMP Dynamic Profile | 434
- Configuring SSM Mapping for Dynamic IGMP and MLD | 436

Dynamic IGMP Configuration Overview

The Internet Group Management Protocol (IGMP) is a host to router signaling protocol for IPv4 used to support IP multicasting. This protocol manages the membership of hosts and routers in multicast groups. IP hosts use IGMP to report their multicast group memberships to any immediately neighboring multicast routers. Multicast routers use IGMP to learn, for each of their attached physical networks, which groups have members.

Subscriber access supports the configuration of IGMP within the `dynamic profiles` hierarchy. By specifying IGMP statements within a dynamic profile, you can dynamically apply IGMP configuration when a subscriber connects to an interface using a particular access technology (DHCP), enabling the subscriber to access a carrier (multicast) network.

Dynamic IGMP consists of a subset of the full range of IGMP capabilities available for static IGMP configuration, applied to dynamic interfaces by means of a dynamic profile. For detailed information about static IGMP configuration, see Configuring IGMP. Much of the static configuration documentation is directly applicable to dynamic IGMP. Note that the following statements that appear in the dynamic IGMP CLI hierarchy are configurable, but have no effect: `accounting`, `group-threshold`, `log-interval`, and `no-accounting`. These statements are not needed at a subscriber level , where typically no more than tens of joins are expected.

Refer to the Multicast Protocols User Guide for a comprehensive understanding of Junos OS support for multicast protocols.

RELATED DOCUMENTATION

Dynamic Profiles Overview

Subscriber Management IGMP Model Overview | 431

Configuring Dynamic DHCP Client Access to a Multicast Network | 432

Configuring IGMP

Subscriber Management IGMP Model Overview

In an IPTV network, channel changes occur when a set-top box (STB) sends IGMP commands that inform an upstream device (for example, a multiservice access node [MSAN] or services router) whether to start or stop sending multicast groups to the subscriber. In addition, IGMP hosts periodically request notification from the STB about which channels (multicast groups) are being received.

You can implement IGMP in the subscriber management network in the following ways:

- **Static IGMP**—All multicast channels are sent to the MSAN. When the MSAN receives an IGMP request to start or stop sending a channel, it adds the subscriber to the multicast group and then discards the IGMP packet.
- **IGMP Proxy**—Only multicast channels currently being viewed are sent to the MSAN. If the MSAN receives a request to view a channel that is not currently being forwarded to the MSAN, it forwards the request upstream. However, the upstream device does not see all channel change requests from each subscriber, limiting bandwidth control options.
- **IGMP Snooping**—Only multicast channels currently being viewed are sent to the MSAN. The MSAN forwards all IGMP requests upstream, unaltered, even if it is already receiving the channel. The upstream device sees all channel change requests from each subscriber. Using IGMP snooping enables the broadband services router to determine the mix of services and the bandwidth requirements of each subscriber and adjust the bandwidth made available to each service.
- **IGMP Passthrough**—The MSAN transparently passes IGMP packets upstream to the broadband services router.

IGMP hosts (sources) also periodically verify that they are sending the correct traffic by requesting that each client send information about what multicast groups it wants to receive. The responses to this *IGMP query* can result in a substantial upstream traffic burst.

IGMPv2 is the minimum level required to support IPTV, and is the most widely deployed. Emerging standards specify IGMPv3.

RELATED DOCUMENTATION

Dynamic IGMP Configuration Overview | 430

Configuring Dynamic DHCP Client Access to a Multicast Network

This topic describes how to create a basic dynamic profile that enables DHCP clients to dynamically access the multicast network.

Before you configure dynamic profiles for initial client access:

1. Create a basic dynamic profile.

   See *Configuring a Basic Dynamic Profile*.

2. Configure the necessary router interfaces that you want accessing DHCP clients to use.

   See *DHCP Subscriber Interface Overview* for information about the types of interfaces you can use with dynamic profiles and how to configure them.

3. Ensure that the router is configured to enable communication between the client and the RADIUS server.

   See *Specifying the Authentication and Accounting Methods for Subscriber Access*.

4. Configure all RADIUS values that you want the profiles to use when validating DHCP clients for access to the multicast network.

   See *RADIUS Servers and Parameters for Subscriber Access*

To configure an initial client access dynamic profile:

1. Access an IGMP access profile.

```
user@host# edit dynamic-profiles access-profile
[edit dynamic-profiles access-profile]
user@host#
```

2. Define the IGMP interface with the interface variable.

NOTE: The variable value is replaced by the name of the interface over which the router received the DHCP message.

```
[edit dynamic-profiles access-profile]
user@host# set protocols igmp interface $junos-interface-name
```

3. (Optional) Enable or disable accounting on the IGMP interface.

```
[edit dynamic-profiles access-profile protocols igmp interface “$junos-interface-name”]
user@host# set accounting
```

or

```
[edit dynamic-profiles access-profile protocols igmp interface “$junos-interface-name”]
user@host# set no-accounting
```

NOTE: This statement enables you to override the accounting setting at the IGMP protocol level. For example, if IGMP accounting is enabled at the `[edit protocols igmp interface interface-name]` hierarchy level, you can use the `no-accounting` statement to disable accounting for any IGMP interfaces that are dynamically created by the dynamic profile. If IGMP accounting is not enabled at the `[edit protocols igmp interface interface-name]` hierarchy level, you can use the `accounting` statement to enable accounting for any IGMP interfaces that are dynamically created by the dynamic profile.

4. Set the IGMP interface to remain enabled.

```
[edit dynamic-profiles access-profile protocols igmp interface “$junos-interface-name”]
user@host# set disable:$junos-igmp-enable
```

NOTE: RADIUS is capable of disabling IGMP. By assigning the enable variable to the `disable` statement, you can ensure that IGMP remains enabled.

5. (Optional) Specify a group policy for the IGMP interface.

```
[edit dynamic-profiles access-profile protocols igmp interface “$junos-interface-name”]
user@host# set group-policy report-reject-policy
```

6. (Optional) Enable immediate leave on the IGMP interface.

```
[edit dynamic-profiles access-profile protocols igmp interface “$junos-interface-name”]
user@host# set immediate-leave:$junos-igmp-immediate-leave
```

7. (Optional) Set the IGMP interface to obtain the IGMP version from RADIUS.

```
[edit dynamic-profiles access-profile protocols igmp interface “$junos-interface-name”]
user@host# set version $junos-igmp-version
```

RELATED DOCUMENTATION

Configuring a Basic Dynamic Profile

Dynamic Profiles Overview

Example: IGMP Dynamic Profile

In this example, IGMP is configured for subscriber access using Junos OS predefined variables.

The predefined variables equate to RADIUS settings as follows:

Junos OS Predefined Variable	RADIUS VSA Name	RADIUS Attribute Number
$var-igmp-version	IGMP-Version	26–78
$var-igmp-access-grp	IGMP-Access-Name	26–71

(Continued)

Junos OS Predefined Variable	RADIUS VSA Name	RADIUS Attribute Number
$var-igmp-access-src-grp	IGMP-Access-Src-Name	26–72

```
[edit dynamic-profiles profile-name]
interfaces {
    demux0 {
        unit "$junos-interface-unit" {
            demux-options {
                underlying-interface "$junos-underlying-interface";
            }
            family inet {
                demux-source {
                    “$junos-subscriber-ip-address”;
                }
                unnumbered-address lo0.0 preferred-source-address 203.0.113.210;
            }
        }
    }
}
protocols {
    igmp {
        interface "$junos-interface-name" {
            version "$var-igmp-version";
            group-policy [ "$var-igmp-access-grp" "$var-igmp-access-src-grp" ];
        }
    }
}
```

NOTE: You must also configure any global IGMP parameters.

RELATED DOCUMENTATION

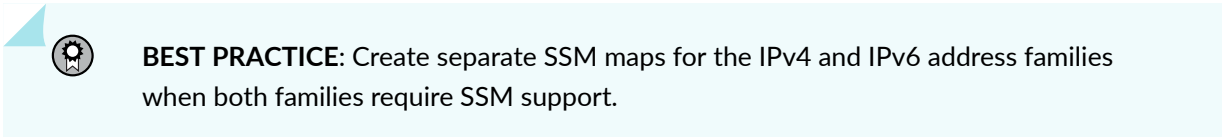
Configuring Dynamic DHCP Client Access to a Multicast Network | 432

Configuring SSM Mapping for Dynamic IGMP and MLD

Source-specific multicast (SSM) is a service model that identifies session traffic by both source and group address. SSM builds shortest-path trees (SPTs) directly represented by (S,G) pairs. The “S” refers to the source's unicast IP address, and the “G” refers to the specific multicast group address. The SSM (S,G) pairs are called channels to differentiate them from any-source multicast (ASM) groups. SSM is ideal for one-to-many multicast services such as network entertainment channels. Although ASM supports one-to-many, its method of source discovery is less efficient than SSM. For example, if you click a link in a browser, ASM notifies the receiver about the group information, but not the source information. With SSM, the client receives both source and group information.

To deploy SSM successfully, you need an end-to-end multicast-enabled network and applications that use an Internet Group Management Protocol version 3 (IGMPv3) or Multicast Listener Discovery version 2 (MLDv2) stack, or you need to configure SSM mapping from IGMPv1 or IGMPv2 to IGMPv3. An IGMPv3 stack provides the capability of a host operating system to use the IGMPv3 protocol.

You can accommodate hosts that do not support IGMPv3 or MLDv1 by using SSM mapping. SSM mapping translates IGMPv1 or IGMPv2 membership reports to an IGMPv3 report, and MLDv1 reports to MLDv2. SSM mapping applies to all group addresses that match the policy, not just those that conform to SSM addressing conventions (232/8 for IPv4, ff30::/32 through ff3F::/32 for IPv6).



BEST PRACTICE: Create separate SSM maps for the IPv4 and IPv6 address families when both families require SSM support.

If you apply an SSM map policy containing both IPv4 and IPv6 addresses to an interface in an IPv4 context (using IGMP), only the IPv4 addresses in the list are used. If there are no such addresses, no action is taken. Similarly, if you apply an SSM map policy containing both IPv4 and IPv6 addresses to an interface in an IPv6 context (using MLD), only the IPv6 addresses in the list are used. If there are no such addresses, no action is taken.

To configure SSM mapping for dynamic IGMP:

1. Create an SSM policy to match the desired IPv4, IPv6, or both group addresses.

```
[edit]
user@host# edit policy-options policy-statement policy-name
```

2. Configure terms for the policy to identify and accept group addresses

```
[edit policy-options policy-statement policy-name]
user@host# set term from name route-filter destination-prefix match-type
user@host# set term name then accept
```

3. Apply the SSM map policy to the dynamic interface in a dynamic profile.

```
[edit dynamic-profiles profile-name protocols (igmp | mld) interface $junos-interface-name]
user@host# set ssm-map-policy ssm-map-policy-name
```

For example, the following configuration creates SSM policy ssm-1. The policy term v4 exactly matches the IPv4 SSM group address 233.252.1.1/32. The policy rejects all other addresses. The policy ssm-1 is then applied to dynamic interfaces created when the igmp-prof dynamic profile is instantiated.

```
[edit]
user@host# edit policy-options policy-statement ssm-1
user@host# set term v4 from route-filter 233.252.1.1/32 exact
user@host# set term v4 then accept
user@host# set then reject
user@host# edit dynamic-profiles mld-prof protocols igmp interface $junos-interface-name
user@host# set ssm-map-policy ssm-1
```

For example, the following configuration creates SSM policy ssm-2. Policy term v6 exactly matches the IPv6 group address ff35::1/128. The policy rejects all other addresses. The policy ssm-2 is then applied to dynamic interfaces created when the mld-prof dynamic profile is instantiated.

```
[edit]
user@host# edit policy-options policy-statement ssm-2
user@host# set term v6 from route-filter ff35::1/128 exact
user@host# set term v6 then accept
user@host# set then reject
user@host# edit dynamic-profiles igmp-prof protocols mld interface $junos-interface-name
user@host# set ssm-map-policy ssm-2
```

RELATED DOCUMENTATION

Dynamic IGMP Configuration Overview | 430

CHAPTER 29

Configuring Dynamic MLD to Enable Subscribers to Access Multicast Networks

IN THIS CHAPTER

- Dynamic MLD Configuration Overview | 439

Dynamic MLD Configuration Overview

The Multicast Listener Discovery (MLD) Protocol manages the membership of hosts and routers in multicast groups. IP version 6 (IPv6) multicast routers use MLD to learn, for each of their attached physical networks, which groups have interested listeners. Each router maintains a list of host multicast addresses that have listeners for each subnet, as well as a timer for each address. However, the router does not need to know the address of the listeners—just the address of the hosts. The router provides addresses to the multicast routing protocol it uses; this ensures that multicast packets are delivered to all subnets where there are interested listeners. In this way, MLD is used as the transport for the Protocol Independent Multicast (PIM) protocol.

Subscriber access supports the configuration of MLD within the `dynamic profiles` hierarchy for dynamically created interfaces. By specifying MLD statements within a dynamic profile, you can dynamically apply MLD configuration when a subscriber connects to an interface using a particular access technology (DHCP), enabling the subscriber to access a carrier (multicast) network.

Dynamic MLD consists of a subset of the full range of MLD capabilities available for static MLD configuration, applied to dynamic interfaces by means of a dynamic profile. For detailed information about static MLD configuration, see Configuring MLD. Much of the static configuration documentation is directly applicable to dynamic MLD. Note that the following statements that appear in the dynamic MLD CLI hierarchy are configurable, but have no effect: `accounting`, `group-threshold`, `log-interval`, and `no-accounting`. These statements are not needed at a subscriber level , where typically no more than tens of joins are expected.

Refer to the Multicast Protocols User Guide for a comprehensive understanding of Junos OS support for multicast protocols.

RELATED DOCUMENTATION

Dynamic Profiles Overview

Configuring Dynamic DHCP Client Access to a Multicast Network | **432**

Configuring MLD

PART 5

Configuring Application-Aware Policy Control and Reporting

- Configuring Application-Aware Policy Control | **442**
- Configuring Application Identification | **468**
- Configuring Reporting for Application-Aware Data Sessions | **479**

CHAPTER 30

Configuring Application-Aware Policy Control

IN THIS CHAPTER

- Understanding Application-Aware Policy Control for Subscriber Management | 442
- Understanding PCC Rules for Subscriber Management | 444
- Configuring Application-Aware Policy Control for Subscriber Management | 446
- Installing Services Packages for Subscriber Management Application-Aware Policy Management | 447
- Configuring Service Data Flow Filters | 448
- Configuring Policy and Charging Control Action Profiles for Subscriber Management | 452
- Configuring Policy and Charging Control Rules | 454
- Configuring a Policy and Charging Control Rulebase | 458
- Configuring a Policy and Charging Enforcement Function Profile for Subscriber Management | 460
- Identifying the Service Interface That Handles Subscriber Management Application-Aware Policy Control | 462
- Configuring PCC Rule Activation in a Subscriber Management Dynamic Profile | 463
- Enabling Direct PCC Rule Activation by a PCRF for Subscriber Management | 465

Understanding Application-Aware Policy Control for Subscriber Management

IN THIS SECTION

- Benefits | 443

Starting in Junos OS Release 16.1R4 and in Junos OS Release 17.2R1, you can configure application-aware policy control, which defines the treatment to apply to a subscriber's packets based on the specific application being used by the subscriber (for example, Facebook) or based on Layer 3 and Layer

4 service data flow (SDF) information for the IP flow (for example, the source and destination IP addresses). Starting in Junos OS Release 19.3R2, application-aware policy control is also supported if you have enabled Next Gen Services on the MX240, MX480 or MX960 router with the MX-SPC3 card.

You configure application-aware policy control by configuring policy and charging control (PCC) rules, which identify the conditions that must be met (such as the application that the traffic is using) and the action to take on that traffic (such as specifying a maximum bit rate). PCC rules can be activated for a subscriber in one of two ways:

- PCC rule activation control by dynamic profile—The dynamic profile assigned to a subscriber identifies a static PCEF profile, which specifies PCC rules. The dynamic profile indicates whether to activate all the rules in the PCEF profile or just a subset of the rules. The PCEF profile and PCC rule names can be variables in the dynamic profile, and the names are obtained by RADIUS during subscriber authorization.
- PCC rule activation by a policy and charging rules function (PCRF) server—Starting in Junos OS Release 18.2R1, a PCRF can directly activate a PCC rule that is configured on the MX Series router by sending a Rule-Install-Name AVP over the Gx interface to the MX Series router during service activation. The specified PCC rule must be identified in a dynamic PCEF profile. If the Rule-Install-Name is also the name of a dynamic profile, then the rule is ignored and the dynamic profile is used.

Benefits

Application-aware policy control allows highly customizable, differentiated services for subscribers.

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
19.3R1	Starting in Junos OS Release 19.3R2, application-aware policy control is also supported if you have enabled Next Gen Services on the MX240, MX480 or MX960 router with the MX-SPC3 card.
18.2R1	Starting in Junos OS Release 18.2R1, a PCRF can directly activate a PCC rule that is configured on the MX Series router by sending a Rule-Install-Name AVP over the Gx interface to the MX Series router during service activation.
16.1R4	Starting in Junos OS Release 16.1R4 and in Junos OS Release 17.2R1, you can configure application-aware policy control, which defines the treatment to apply to a subscriber's packets based on the specific application being used by the subscriber (for example, Facebook) or based on Layer 3 and Layer 4 service data flow (SDF) information for the IP flow (for example, the source and destination IP addresses).

RELATED DOCUMENTATION

Understanding PCC Rules for Subscriber Management | 444
Configuring Application-Aware Policy Control for Subscriber Management | 446
Configuring Policy and Charging Control Action Profiles for Subscriber Management | 452
Configuring Service Data Flow Filters | 448
Configuring Policy and Charging Control Rules | 454
Application Identification Overview | 468

Understanding PCC Rules for Subscriber Management

IN THIS SECTION

- Application Filters | 444
- Service Data Flow Filters | 445
- PCC Action Profiles | 445

NOTE: Starting in Junos OS Release 19.3R2, PCC rules are also supported if you have enabled Next Gen Services on the MX240, MX480 or MX960 router with the MX-SPC3 card.

Policy and charging control (PCC) rules define the treatment to apply to subscriber traffic based on the application being used by the subscriber (for example, Facebook) or based on the Layer 3 and Layer 4 service data flow (SDF) information for the IP flow (for example, the source and destination IP addresses). You configure PCC rules, and PCC rules are then activated by either the subscriber's dynamic profile or by a PCRF. PCC rules include the following components:

Application Filters

Applications and application groups are specified in the `from` clause of a PCC rule to identify IP packets belonging to a specific application. If the IP packet is for an application identified in a PCC rule, the treatment specified in the PCC action profile in the `then` clause of the rule is applied.

To configure application-aware PCC rules, you can specify one or more of the following parameters:

- application—Specifies the name of an application. This can be a Layer 7 protocol (for example, HTTP) or a particular application running on a Layer 7 protocol, such as Facebook and Yahoo Messenger.
- application-group—Specifies the name of an application group, which represents a collection of Layer 7 applications that can be processed at the same time.

NOTE: Application-aware PCC rules that reference specified applications can include wildcard or specific Layer-3 SDF filters, Layer-4 SDF filters, or both.

You can see a list of all the applications and application groups by using the `show services application-identification application` command. To configure a custom application, see "Configuring Custom Application Signatures" on page 472.

Service Data Flow Filters

SDF filters (flow identifiers) are specified in the `from` clause of a PCC rule to identify IP packets belonging to a particular Layer 3 or Layer 4 service data flow. If the IP packet matches the SDF filter in a PCC rule, the treatment specified in the PCC action profile in the `then` clause of the rule is applied.

To configure Layer 3 or Layer 4 SDF filters, you specify one or more of the following parameters:

- Source IP address
- Destination IP address
- Source port
- Destination port
- Layer 4 protocol (UDP or TCP)

PCC Action Profiles

A PCC rule configuration includes an action profile in the `then` clause that defines the treatment to apply to a packet belonging to an application or to an SDF identified in the `from` clause of the rule. You can configure a PCC action profile that is used in one or more PCC rules to provide the following functionality:

- HTTP redirection—Specifies HTTP redirection to a URL. You can use this action only for PCC rules that match only HTTP-based applications and all flows.
- HTTP Steering path—Specifies an IPv4 or IPv6 address for steering HTTP packets. You can use this action only for PCC rules that match only HTTP-based applications and all flows.

NOTE: A single PCC rule can support either HTTP redirection or HTTP steering path, but not both.

- Steering with a routing instance—Specifies a routing instance for steering of packets. You can configure different routing instances for traffic from the subscriber (uplink) and traffic to the subscriber (downlink).
- Forwarding class—Specifies the forwarding class that you want assigned to the packet.
- Maximum bit rate—Specifies the maximum bit rate for uplink and for downlink traffic.
- Gating status—Specifies whether to block or to forward IP packets.

RELATED DOCUMENTATION

Configuring Application-Aware Policy Control for Subscriber Management | 446

Configuring Policy and Charging Control Action Profiles for Subscriber Management | 452

Configuring Service Data Flow Filters | 448

Configuring Policy and Charging Control Rules | 454

Configuring Application-Aware Policy Control for Subscriber Management

This topic gives an overview of the tasks you perform to configure policy control for subscriber management based on the layer 7 application that traffic is using or on the particular Layer 3 or Layer 4 service data flow.

NOTE: Starting in Junos OS Release 19.3R2, application-aware policy control is also supported if you have enabled Next Gen Services on the MX240, MX480 or MX960 router with the MX-SPC3 card.

To configure policy control:

1. Install service packages on any MS-MPC PICs that perform application-aware policy control, or on the MX-SPC3 services card if you have enabled Next Gen Services on either the MX240, MX480, or MX860.

   See "Installing Services Packages for Subscriber Management Application-Aware Policy Management" on page 447.

2. Configure any service data flow filters to be used in PCC rules.

   See "Configuring Service Data Flow Filters" on page 448.

3. Configure any custom applications to be used in PCC rules.

   See "Configuring Custom Application Signatures" on page 472.

4. Configure the PCC action profiles to be used in PCC rules.

   See "Configuring Policy and Charging Control Action Profiles for Subscriber Management" on page 452

5. Configure PCC rules.

   See "Configuring Policy and Charging Control Rules" on page 454.

6. (Optional) Configure PCC rulebases.

   See "Configuring a Policy and Charging Control Rulebase" on page 458.

7. Configure a policy and charging enforcement function (PCEF) profile.

   See "Configuring a Policy and Charging Enforcement Function Profile for Subscriber Management" on page 460.

8. Configure a service set for application-aware policy control.

   See "Identifying the Service Interface That Handles Subscriber Management Application-Aware Policy Control" on page 462.

9. Perform one of the following:

   - For PCC rule activation through a dynamic profile, perform "Configuring PCC Rule Activation in a Subscriber Management Dynamic Profile" on page 463.
   - For direct PCC rule activation by a policy and charging rules function (PCRF) server, perform "Enabling Direct PCC Rule Activation by a PCRF for Subscriber Management" on page 465.

RELATED DOCUMENTATION

Understanding Application-Aware Policy Control for Subscriber Management | 442

Installing Services Packages for Subscriber Management Application-Aware Policy Management

You must install a set of service packages on any MS-MPC PICs that perform application-aware policy control, or on the MX-SPC3 services card if you have enabled Next Gen Services on the MX240, MX480, or MX960.

To install service packages:

1. Specify the MS-MPC PIC or MX-SPC3 services card.

```
[edit chassis]
user@host# edit fpc slot-number pic pic-number
```

2. Install the services packages.

```
[edit chassis fpc slot-number pic pic-number adaptive-services service-package extension-
provider ]
user@host# set package jservices-mss
user@host# set package jservices-jdpi
user@host# set package jservices-pcef
```

RELATED DOCUMENTATION

Understanding Application-Aware Policy Control for Subscriber Management | 442

Configuring Service Data Flow Filters

NOTE: Starting in Junos OS Release 19.3R2, PCC rules are also supported if you have enabled Next Gen Services on the MX240, MX480 or MX960 router with the MX-SPC3 card.

A service data flow (SDF) filter is specified as a matching condition in the `from` clause of a policy and charging control (PCC) rule. Each SDF filter can have one or more flows associated with it; each flow is a five-tuple match.

NOTE: If you configure an SDF filter without specifying a remote address, port, port range, or protocol , then the SDF filter matches IP packets that have any value configured for the corresponding attribute. If you configure an SDF filter, you must configure at least one of the following attributes: direction, local port or local port range, protocol, remote address, or remote port or remote port range.

You can configure SDF filters for Junos OS Subscriber Aware or for Junos OS Broadband Subscriber Management, but you use a different CLI hierarchy level for each product.

- If you are using Junos OS Subscriber Aware, configure SDF filters at the `[edit unified-edge pcef]` hierarchy level.
- If you are using Junos OS Broadband Subscriber Management, configure SDF filters at the `[edit services pcef]` hierarchy level.

To configure Layer 3 and Layer 4 SDF filters:

1. Specify a name for the SDF filter.

   For Junos OS Subscriber Aware:

   ```
   [edit unified-edge pcef]
   user@host# set flow-descriptions flow-identifier
   ```

   For Junos OS Broadband Subscriber Management:

   ```
   [edit services pcef]
   user@host# set flow-descriptions flow-identifier
   ```

2. Specify the flow direction for the SDF filter.

   NOTE: If you do not specify a flow direction, then the SDF filter is applied in both the uplink and downlink directions.

   For Junos OS Subscriber Aware:

   ```
   [edit unified-edge pcef flow-descriptions flow-identifier]
   user@host# set direction (uplink | downlink | both)
   ```

   For Junos OS Broadband Subscriber Management:

   ```
   [edit services pcef flow-descriptions flow-identifier]
   user@host# set direction (uplink | downlink | both)
   ```

3. Specify a remote address (IPv4 or IPv6) for the SDF filter:

   NOTE: You can specify an IPv4 subnet or an IPv6 subnet but not both.

   - Specify an IPv4 address for the SDF filter.

For Junos OS Subscriber Aware:

```
[edit unified-edge pcef flow-descriptions flow-identifier]
user@host# set remote-address ipv4-address ipv4-address
```

For Junos OS Broadband Subscriber Management:

```
[edit services pcef flow-descriptions flow-identifier]
user@host# set remote-address ipv4-address ipv4-address
```

- Specify an IPv6 address for the SDF filter.

  For Junos OS Subscriber Aware:

  ```
  [edit unified-edge pcef flow-descriptions flow-identifier]
  user@host# set remote-address ipv6-address ipv6-address
  ```

  For Junos OS Broadband Subscriber Management:

  ```
  [edit services pcef flow-descriptions flow-identifier]
  user@host# set remote-address ipv6-address ipv6-address
  ```

4. Specify a protocol (using the standard protocol number) for the SDF filter.

   For Junos OS Subscriber Aware:

   ```
   [edit unified-edge pcef flow-descriptions flow-identifier]
   user@host# set protocol number
   ```

   For Junos OS Broadband Subscriber Management:

   ```
   [edit services pcef flow-descriptions flow-identifier]
   user@host# set protocol number
   ```

5. Specify a local port or a list of port numbers for the SDF filter. To specify a list of port numbers (up to a maximum of three), enclose the port numbers in square brackets ([]).

NOTE: You can configure a local port or local port range but not both in the same SDF filter.

For Junos OS Subscriber Aware:

```
edit unified-edge pcef flow-descriptions flow-identifier]
user@host# set local-ports number
```

For Junos OS Broadband Subscriber Management:

```
edit services pcef flow-descriptions flow-identifier]
user@host# set local-ports number
```

6. Specify a local port range for the SDF filter.

   For Junos OS Subscriber Aware:

```
[edit unified-edge pcef flow-descriptions flow-identifier]
user@host# set local-port-range low low-value high high-value
```

   For Junos OS Broadband Subscriber Management:

```
[edit services pcef flow-descriptions flow-identifier]
user@host# set local-port-range low low-value high high-value
```

7. Specify a remote port or list of remote ports for the SDF filter. To specify a list of port numbers (up to a maximum of three), enclose the port numbers in square brackets ([]).

NOTE: You can configure a remote port or remote port range but not both in the same SDF filter.

For Junos OS Subscriber Aware:

```
[edit unified-edge pcef flow-descriptions flow-identifier]
user@host# set remote-ports number
```

For Junos OS Broadband Subscriber Management:

```
[edit services pcef flow-descriptions flow-identifier]
user@host# set remote-ports number
```

8. Specify a remote port range for the SDF filter.

   For Junos OS Subscriber Aware:

```
[edit unified-edge pcef flow-descriptions flow-identifier]
user@host# set remote-port-range low low-value high high-value
```

   For Junos OS Broadband Subscriber Management:

```
[edit services pcef flow-descriptions flow-identifier]
user@host# set remote-port-range low low-value high high-value
```

RELATED DOCUMENTATION

- Configuring Application-Aware Policy Control for Subscriber Management | 446
- Understanding Application-Aware Policy Control for Subscriber Management | 442
- *Understanding Predefined Policy and Charging Control Rules for Subscriber-Aware Traffic Treatment*
- *Configuring Policy and Charging Control Rules*

Configuring Policy and Charging Control Action Profiles for Subscriber Management

A PCC action profile defines the treatment to be applied to a subscriber's packets associated with specific applications or with specific service data flows. A PCC action profile is specified in the `then` clause of a PCC rule.

NOTE: You cannot change a PCC action profile while it is being used by a subscriber. To modify the PCC action profile, you must log off the subscribers that are using the PCC rule that includes the profile.

To configure PCC action profiles:

1. Specify a name for the PCC action profile.

```
[edit services pcef]
user@host# edit pcc-action-profiles profile-name
```

2. Configure the maximum bit rate for uplink and downlink subscriber traffic.

```
[edit services pcef pcc-action-profiles profile-name]
user@host# set maximum-bit-rate uplink mbr-uplink-value downlink mbr-downlink-value
```

   The range is 0 through 6144000 Kbps.

3. Configure HTTP redirection to a URL.

```
[edit services pcef pcc-action-profiles profile-name redirect]
user@host# set url url-name
```

   NOTE: A PCC action profile that includes HTTP redirection can only be used in PCC rules that match only HTTP-based applications and all flows.

4. Configure the steering of traffic to a third-party server for applying services or to a service chain with one of the following methods:

   - Specify the IP address of the third-party server for HTTP traffic.

```
[edit services pcef pcc-action-profiles profile-name ]
user@host# set steering path (ipv4-address ipv4-address | set ipv6-address ipv6-address)
```

     NOTE: A PCC action profile that includes a steering path can only be used in PCC rules that match only HTTP-based applications and all flows.

   - Specify the routing instance to use to reach the third-party server or service chain.

```
[edit services pcef pcc-action-profiles profile-name]
user@host# set steering routing-instance downlink downlink-vrf-name uplink uplink-vrf-name
```

     The downlink routing instance is applied to traffic going to the access side, and the uplink routing instance is applied to traffic being sent from the access side.

5. Specify that the PCC action profile steering attributes that a PCC rule applies at the start of a data flow will continue to be applied to that data flow when the PCC rule match conditions are modified, deleted, or added to.

```
[edit services pcef pcc-action-profiles profile-name steering]
user@host# set keep-existing-steering
```

6. Specify the forwarding class to assign to packets.

```
[edit services pcef pcc-action-profiles profile-name]
user@host# set forwarding-class class-name
```

7. Configure the gating status by enabling or disabling the forwarding of packets.

```
[edit services pcef pcc-action-profiles profile-name]
user@host# set gate-status (disable-both | downlink | uplink | uplink-downlink)
```

RELATED DOCUMENTATION

- Configuring Application-Aware Policy Control for Subscriber Management | 446
- Understanding Application-Aware Policy Control for Subscriber Management | 442
- Configuring Policy and Charging Control Rules | 454

Configuring Policy and Charging Control Rules

A policy and charging control (PCC) rule defines the treatment to be applied to packets associated with specific applications or to specific service data flows.

You can configure PCC rules for Junos OS Subscriber Aware or for Junos OS Broadband Subscriber Management, but you use a different CLI hierarchy level for each product.

- If you are using Junos OS Subscriber Aware, configure PCC rules at the `[edit unified-edge pcef]` hierarchy level.
- If you are using Junos OS Broadband Subscriber Management, configure PCC rules at the `[edit services pcef]` hierarchy level.

NOTE: If you are using Junos OS Subscriber Aware, you must be in maintenance mode to make a change to a PCC rule. (See Changing PCEF Profiles, PCC Rules, PCC Rulebases, Diameter Profiles, Flow Descriptions, and PCC Action Profiles).

NOTE: If you are using Junos OS Broadband Subscriber Management, you cannot change a PCC rule while it is being used by a subscriber. To modify the rule, you must log off the subscribers that are using the rule.

Before you configure PCC rules, you must do the following:

- Configure the service data flow (SDF) filters that the PCC rules reference.
- Configure the application groups and any custom applications that you want to reference in application-aware PCC rules.
- Configure the PCC action profiles that the PCC rules reference.

NOTE: When specifying application-aware PCC rules in a PCEF profile, you must also configure a default Layer 3 or Layer 4 wildcard PCC rule to ensure that the default charging characteristics are applied to unmatched subscriber traffic without dropping that traffic. For example, the default Layer 3 or Layer 4 wildcard PCC rule prevents traffic based on DNS queries from being dropped. In addition, the policy (PCEF profile) that includes application-aware PCC rules must also include a wildcard Layer 3 or Layer 4 PCC rule at a lower precedence.

To configure PCC rules:

1. Specify a name for the PCC rule.

   For Junos OS Subscriber Aware:

   ```
   [edit unified-edge pcef]
   user@host# edit pcc-rules rule-name
   ```

   For Junos OS Broadband Subscriber Management:

   ```
   [edit services pcef]
   user@host# edit pcc-rules rule-name
   ```

2. In a `from` statement, specify an SDF filter to use Layer 3 or Layer 4 match conditions for filtering subscriber traffic.

   For Junos OS Subscriber Aware:

   ```
   [edit unified-edge pcef pcc-rules  rule-name]
   user@host# set from flows flow-identifier
   ```

   For Junos OS Broadband Subscriber Management:

   ```
   [edit services pcef pcc-rules  rule-name]
   user@host# set from flows flow-identifier
   ```

   If you do not want to filter subscriber traffic based on SDF filters, use the `any` option.

   For Junos OS Subscriber Aware:

   ```
   [edit unified-edge pcef pcc-rules  rule-name]
   user@host# set from flows any
   ```

   For Junos OS Broadband Subscriber Management:

   ```
   [edit services pcef pcc-rules  rule-name]
   user@host# set from flows any
   ```

3. (Optional) Specify an application as a match condition for filtering subscriber traffic.

   For Junos OS Subscriber Aware:

   ```
   [edit unified-edge pcef pcc-rules  rule-name]
   user@host# set from applications application-name
   ```

   For Junos OS Broadband Subscriber Management:

   ```
   [edit services pcef pcc-rules  rule-name]
   user@host# set from applications application-name
   ```

4. (Optional) Specify multiple applications instead of specifying each application separately by specifying an application group as a match condition for filtering subscriber traffic.

For Junos OS Subscriber Aware:

```
[edit unified-edge pcef pcc-rules rule-name]
user@host# set from application-groups application-group-name
```

For Junos OS Broadband Subscriber Management:

```
[edit services pcef pcc-rules rule-name]
user@host# set from application-groups application-group-name
```

5. Specify the PCC rules action profile that defines the treatment to be applied to specific service data flows or to packets associated with specific applications.

> **NOTE**: You can use PCC action profiles with HTTP redirection or HCM profiles only in PCC rules that match only HTTP-based applications and any flows.

For Junos OS Subscriber Aware:

```
[edit unified-edge pcef pcc-rules rule-name]
user@host# set then pcc-action-profile profile-name
```

For Junos OS Broadband Subscriber Management:

```
[edit services pcef pcc-rules rule-name]
user@host# set then pcc-action-profile profile-name
```

RELATED DOCUMENTATION

Understanding Predefined Policy and Charging Control Rules for Subscriber-Aware Traffic Treatment

Understanding Application-Aware Policy Control for Subscriber Management | **442**

Configuring Policy and Charging Control Rules

Configuring Policy and Charging Control Action Profiles for Subscriber Management | **452**

Configuring Service Data Flow Filters

Configuring Custom Application Signatures

Configuring a Policy and Charging Control Rulebase

A policy and charging control (PCC) rulebase contains a set of PCC rules. Each rule specified in the PCC rulebase is assigned a precedence to designate the priority in which PCC rules are evaluated for selection in a policy and charging enforcement function (PCEF) profile.

NOTE: Starting in Junos OS Release 19.3R1, application-aware policy control is also supported for Broadband Subscriber Management if you have enabled Next Gen Services on the MX240, MX480 or MX960 router with the MX-SPC3 card.

You can configure PCC rulebases for Junos OS Subscriber Aware or for Junos OS Broadband Subscriber Management, but you use a different CLI hierarchy level for each product.

- If you are using Junos OS Subscriber Aware, configure PCC rulebases at the `[edit unified-edge pcef]` hierarchy level.
- If you are using Junos OS Broadband Subscriber Management, configure PCC rulebases at the `[edit services pcef]` hierarchy level.

NOTE: If you are using Junos OS Subscriber Aware, you must be in maintenance mode to make a change to a PCC rulebase. (See Changing PCEF Profiles, PCC Rules, PCC Rulebases, Diameter Profiles, Flow Descriptions, and PCC Action Profiles).

NOTE: If you are using Junos OS Broadband Subscriber Management, you cannot change a PCC rulebase while it is being used by a subscriber. To modify the rulebase, you must log off the subscribers that are using the rule.

Before you configure a PCC rulebase, you must do the following:

- Configure service data flow filters.
- Configure PCC action profiles.
- Configure PCC rules.

To configure a PCC rulebase:

1. Specify a name for the rulebase.

For Junos OS Subscriber Aware:

```
[edit unified-edge pcef ]
user@host#  edit pcc-rulebases rulebase-name
```

For Junos OS Broadband Subscriber Management:

```
[edit services pcef ]
user@host#  edit pcc-rulebases rulebase-name
```

2. Specify the PCC rules that the rulebase references and a precedence value (1 through 65,535) for each rule.

NOTE:

- The same rule can be configured in different rulebases and can have a different precedence.
- The precedence assigned must be unique among the configured PCC rules.
- A lower precedence value indicates a higher precedence. For example, if a PCC rulebase has two PCC rules with precedence 5 and 10 respectively, the PCC rule with precedence 5 is evaluated first.

For Junos OS Subscriber Aware:

```
[edit unified-edge pcef pcc-rulebases rulebase-name]
user@host#  set pcc-rule rule-name precedence number
user@host#  set pcc-rule rule-name precedence number
user@host#  set pcc-rule rule-name precedence number
```

For Junos OS Broadband Subscriber Management:

```
[edit services pcef pcc-rulebases rulebase-name]
user@host#  set pcc-rule rule-name precedence number
user@host#  set pcc-rule rule-name precedence number
user@host#  set pcc-rule rule-name precedence number
```

RELATED DOCUMENTATION

Configuring Policy and Charging Control Rules

Understanding Predefined Policy and Charging Control Rules for Subscriber-Aware Traffic Treatment

Understanding Application-Aware Policy Control for Subscriber Management | 442

Understanding Junos Subscriber Aware Policy and Charging Enforcement Function (PCEF)

Configuring a Policy and Charging Enforcement Function Profile for Subscriber Management

NOTE: Starting in Junos OS Release 19.3R2, policy and charging enforcement function (PCEF) profiles are also supported if you have enabled Next Gen Services on the MX240, MX480 or MX960 router with the MX-SPC3 card.

A PCEF profile specifies a set of PCC rules and rulebases that can be assigned to a subscriber, and assigns a precedence value to each predefined rule. The PCEF profile is used in one of the following ways:

- A static PCEF profile is specified in a dynamic profile. The dynamic profile indicates whether to activate all the rules in the PCEF profile or just a subset of the rules.
- A dynamic PCEF profile identifies the PCC rules and rulebases that a PCRF can directly activate.

NOTE: You cannot change a PCEF profile while it is being used by a subscriber. To modify the PCEF profile, you must log off the subscribers that are using the PCEF profile.

To configure a PCEF profile:

1. Specify a name for the PCEF profile.

```
[edit services  pcef]
user@host#  edit profiles profile-name
```

2. Specify one or more PCC rules and a precedence for each rule. A lower precedence value indicates a higher precedence. The precedence assigned must be unique among the configured PCC rules, including the PCC rules that are assigned a precedence within a PCC rulebase.

- For a PCEF profile that is specified in a dynamic profile, specify the rules under static-policy-control.

```
[edit services  pcef profiles profile-name]
user@host#  set static-policy-control pcc-rules rule-name precedence number
```

- For a PCEF profile that identifies the PCC rules that a PCRF can directly activate, specify the rules under dynamic-policy-control.

```
[edit services  pcef profiles profile-name]
user@host#  set dynamic-policy-control pcc-rules rule-name precedence number
```

3. Specify one or more PCC rulebases.

- For a PCEF profile that is specified in a dynamic profile, specify the rulebases under static-policy-control.

```
[edit services  pcef profiles profile-name]
user@host#  set static-policy-control pcc-rulebases rulebase-name
```

- For a PCEF profile that identifies the PCC rules that a PCRF can directly activate, specify the rulebases under dynamic-policy-control.

```
[edit services pcef profiles profile-name]
user@host# set dynamic-policy-control pcc-rulebases rulebase-name
```

RELATED DOCUMENTATION

Configuring Application-Aware Policy Control for Subscriber Management | 446

Configuring Policy and Charging Control Rules | 454

Configuring a Policy and Charging Control Rulebase | 458

Identifying the Service Interface That Handles Subscriber Management Application-Aware Policy Control

NOTE: Starting in Junos OS Release 19.3R2, application-aware policy control is also supported if you have enabled Next Gen Services on the MX240, MX480 or MX960 router with the MX-SPC3 card.

Configure a service set to identify the service interface that handles application-aware policy control.

To configure a service set for application-aware policy control:

1. Define an application-aware service set.

```
[edit services]
user@host# set service-set service-set-name service-set-options subscriber-awareness
```

2. Enable PCEF services for the service set by specifying a dummy name for the `pcef-profile`.

   a. Configure a dummy PCEF profile.

```
[edit services pcef]
user@host# set profiles profile-name
```

   b. Specify the dummy profile in the service set.

```
[edit services service-set service-set-name]
user@host# set pcef-profile pcef-profile-name
```

3. Enable application identification for the service set by specifying a dummy name for the `application-identification-profile`.

   a. Configure a dummy application identification profile.

```
[edit services application-identification]
user@host# set profile app-id-profile-name
```

b. Specify the dummy profile in the service set.

```
[edit services service-set service-set-name]
user@host# set application-identification-profile app-id-profile-name
```

4. Specify the services PIC interface on which the services are performed.

```
[edit services service-set service-set-name]
user@host# set interface-service service-interface interface-name
```

If you have redundancy configured, the *interface-name* is ams*n* if you do not have Next Gen Services enabled, and is ams0.1 if you have enabled Next Gen Services on the MX240, MX480 or MX960 router with the MX-SPC3 card.

If you do not have redundancy configured, the *interface-name* is ms-*fpc*/*pci*/0 if you do not have Next Gen Services enabled on the MX-SPC3 services card on the MX240, MX480, or MX860 router, and is vsp-*fpc*/*pci*/0 if you do have Next Gen Services enabled for MX-SPC3 services card on the MX router.

RELATED DOCUMENTATION

Configuring Application-Aware Policy Control for Subscriber Management | 446

Configuring PCC Rule Activation in a Subscriber Management Dynamic Profile

NOTE: Starting in Junos OS Release 19.3R2, PCC rule activation in a dynamic profile is also supported if you have enabled Next Gen Services on the MX240, MX480 or MX960 router with the MX-SPC3 card.

To configure PCC rule activation by a dynamic profile, specify the PCEF profile to use, the PCC rules to activate, and the service set to use.

1. Assign a PCEF profile to the dynamic profile. In the client dynamic profile, you can identify the PCEF profile with the variable `$junos-pcef-profile`. All of a subscriber's dynamic profiles that include a PCEF profile must point to the same PCEF profile.

```
[edit dynamic-profile profile-name interfaces interface-name unit logical-unit-number service]
user@host# set pcef pcef-profile-name
```

2. Activate PCC rules in the dynamic profile. In the access profile, you can identify a rule name with the variable `$junos-pcef-rule`.

NOTE: Do not activate both service data flow (Layer 3 or Layer 4) PCC rules that have a gating action and application-aware (Layer 7) PCC rules in the same dynamic profile. The gating action for the service data flow PCC rules is not applied in this situation.

To activate one PCC rule:

```
[edit dynamic-profile profile-name interfaces interface-name unit logical-unit-number service
pcef pcef-profile-name]
user@host# set activate rule-name
```

To activate all the PCC rules:

```
[edit dynamic-profile profile-name interfaces interface-name unit logical-unit-number service
pcef pcef-profile-name]
user@host# set activate-all
```

3. Assign a service set to the dynamic profile. This must be a service set that you configured for application-aware policy control. In the client dynamic profile, you can identify the service set with a variable (`$junos-input-service-set` | `$junos-output-service-set` | `$junos-input-ipv6-service-set` | `$junos-output-ipv6-service-set`). You must use the same service set for both the input and output service.

```
[edit dynamic-profile profile-name interfaces interface-name unit logical-unit-number family
family service (input | output) service-set]
user@host# set service-set service-set-name
```

4. (Optional) Assign a service filter to the dynamic profile. The service filter can identify conditions for which you want to skip application-aware policy control. In the client dynamic profile, you can

identify the service filter with a variable (`$junos-input-service-filter` | `$junos-output-service-filter` | `$junos-input-ipv6-service-filter` | `$junos-output-ipv6-service-filter`).

```
[edit dynamic-profile profile-name interfaces interface-name unit logical-unit-number family
family service (input | output) service-set service-set-name]
user@host# set service-filter filter-name
```

RELATED DOCUMENTATION

Configuring Application-Aware Policy Control for Subscriber Management | 446

Understanding Application-Aware Policy Control for Subscriber Management | 442

Enabling Direct PCC Rule Activation by a PCRF for Subscriber Management

NOTE: Starting in Junos OS Release 19.3R1, direct PCC rule activation by a PCRF is also supported if you have enabled Next Gen Services on the MX240, MX480 or MX960 router with the MX-SPC3 card.

Enable direct PCRF activation of PCC rules by configuring a PCRF partition, a Diameter instance, and a PCC context in an access profile.

1. Configure the Diameter instance. See *Configuring Diameter*.
2. Configure the PCRF partition. See *Configuring the PCRF Partition*.
3. Enable PCRF provisioning in the access profile.

```
[edit access profile profile-name]
user@host# set provisioning-order pcrf
```

4. Assign a PCEF profile to the access profile PCC context.

```
[edit access profile profile-name session-options pcc-context]
user@host# set profile-name pcef-profile-name
```

5. Specify the IPv6 input service set that handles application-aware policy control.

```
[edit access profile profile-name session-options pcc-context]
user@host# set ipv6-input-service-set-name service-set-name
```

6. (Optional) Specify a service filter for the IPv6 input service set to identify conditions for which you want to skip application-aware policy control.

```
[edit access profile profile-name session-options pcc-context]
user@host# set ipv6-input-service-filter-name filter-name
```

7. Specify the IPv4 input service set that handles application-aware policy control.

```
[edit access profile profile-name session-options pcc-context]
user@host# set input-service-set-name service-set-name
```

8. (Optional) Specify a service filter for the IPv4 input service set to identify conditions for which you want to skip application-aware policy control.

```
[edit access profile profile-name session-options pcc-context]
user@host# set input-service-filter-name filter-name
```

9. Specify the IPv6 output service set that handles application-aware policy control.

```
[edit access profile profile-name session-options pcc-context]
user@host# set ipv6-output-service-set-name service-set-name
```

10. (Optional) Specify a service filter for the IPv6 output service set to identify conditions for which you want to skip application-aware policy control.

```
[edit access profile profile-name session-options pcc-context]
user@host# set ipv6-output-service-filter-name filter-name
```

11. Specify the IPv4 output service set that handles application-aware policy control.

```
[edit access profile profile-name session-options pcc-context]
user@host# set output-service-set-name service-set-name
```

12. (Optional) Specify a service filter for the IPv4 output service set to identify conditions for which you want to skip application-aware policy control.

```
[edit access profile profile-name session-options pcc-context]
user@host# set output-service-filter-name filter-name
```

RELATED DOCUMENTATION

Configuring Application-Aware Policy Control for Subscriber Management | 446

Understanding Application-Aware Policy Control for Subscriber Management | 442

CHAPTER 31

Configuring Application Identification

IN THIS CHAPTER

- Application Identification Overview | 468
- Downloading and Installing Predefined Junos OS Application Signature Packages | 470
- Improving the Application Traffic Throughput | 471
- Configuring Custom Application Signatures | 472
- Uninstalling a Predefined Junos OS Application Signature Package | 478

Application Identification Overview

Junos Application Aware is an infrastructure plug-in on MS-MPC service PICs and on the MX-SPC3 services card that provides information to clients about application protocol bundles based on deep packet inspection (DPI) of application signatures. These clients can be any of the plug-ins on the MX Series router service chain, such as traffic detection function (TDF), that request application classification data. Starting in Junos OS Release 16.1R4 and Junos OS Release 17.2R1, application identification is available in Junos OS Broadband Subscriber Management. Starting in Junos OS Release 19.3R2, application identification is also supported for Broadband Subscriber Management on the MX-SPC3 services card if you have enabled Next Gen Services on the MX240, MX480 or MX960 router.

In application identification, you can apply application signatures as follows:

- **Predefined signatures**—Junos Application Aware comes with a bundle of predefined, preinstalled application signatures, but we recommend that you download and install the latest version of predefined signatures. As new sets of signatures are supported, they are compiled and made available for you to download.

- **Custom application signatures**—For any application signatures that are not predefined, you can create custom signatures for HTTP, SSL, and stream signature contexts and install them for application identification. After you have configured and committed custom signatures, they are serialized and merged with the predefined application signatures. You can specify the following types of custom application signatures:

- **Address based**—You can define an application identification based on a specific IP address, or port, or both where a source IP address, destination IP address, or both are used for a known application in a customer's network. This is useful, for example, when a Session Initiation Protocol (SIP) server initiates a session from its well known port, 5060. The customer can put the SIP server IP address and port 5060 as source IP/port for the SIP application. This method provides efficiency and accuracy of application identification for customer's network.
- **Internet Control Message Protocol (ICMP) based**—Application identification based on types of ICMP messages.
- **IP protocol based**—Application identification based on IP protocol. TCP, UDP, and ICMP are not supported for this method of signature creation.
- **Pattern-matching signatures**—Application based on pattern matching combined with Layer 7 protocol identification.

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
19.3R1	Starting in Junos OS Release 19.3R2, application identification is also supported for Broadband Subscriber Management on the MX-SPC3 services card if you have enabled Next Gen Services on the MX240, MX480 or MX960 router.
16.1R4	Starting in Junos OS Release 16.1R4 and Junos OS Release 17.2R1, application identification is available in Junos OS Broadband Subscriber Management.

RELATED DOCUMENTATION

Configuring Custom Application Signatures

Downloading and Installing Predefined Junos OS Application Signature Packages

Downloading and Installing Predefined Junos OS Application Signature Packages

NOTE: Starting in Junos OS Release 19.3R2 and 19.4R1, application identification is also supported for Broadband Subscriber Management if you have enabled Next Gen Services on the MX240, MX480 or MX960 router with the MX-SPC3 card.

To download, install, and verify the installation of predefined Junos OS application signature packages:

1. Use `download ignore-server-validation` if you want to skip server certification validation during the download. Validation is enabled by default.

```
[edit services application-identification]
user@host# set download ignore-server-validation
```

2. Configure the URL for the application signature packages server.

```
[edit services application-identification]
user@host# set download url https://signatures.juniper.net/cgi-bin/index.cgi
```

3. Download the application signature package.
   - To download the latest signature package, enter the following command:

```
user@host> request services application-identification download
```

   - To download a specific, known signature package, include the version number:

```
user@host> request services application-identification download version version-number
```

4. Confirm the successful download of the package.

```
user@host> request services application-identification download status
```

```
Downloading application package succeed.
```

5. Install the application signature package.

```
user@host> request services application-identification install
```

6. Confirm the successful installation of the application signature package.

```
user@host> request services application-identification install status
```

```
Compiling application signatures of package version.
```

or

```
Install application package succeed
```

7. View the protocol bundle status:

```
user@host> show services application-identification status
```

RELATED DOCUMENTATION

Uninstalling a Predefined Junos OS Application Signature Package

Application Identification Overview

Configuring Custom Application Signatures

Improving the Application Traffic Throughput

NOTE: Starting in Junos OS Release 19.3R2, application identification is also supported for Broadband Subscriber Management if you have enabled Next Gen Services on the MX240, MX480 or MX960 router with the MX-SPC3 card.

To improve the application throughput when using application identification for deep packet inspection (DPI):

Enable the DPI performance mode.

```
[edit services application-identification]
user@host# set enable-performance-mode
```

This limits the maximum DPI processing to four packets per session.

By default, DPI performance mode is disabled.

Configuring Custom Application Signatures

NOTE: Starting in Junos OS Release 19.3R2 and 19.4R1, application identification is also supported for Broadband Subscriber Management if you have enabled Next Gen Services on the MX240, MX480 or MX960 router with the MX-SPC3 card.

You can configure custom application definitions using custom signatures. These definitions enable identification of protocol bundles through deep packet inspection (DPI) for use by interested services in the service chain.

Before you configure custom application signatures, ensure that `jservices-jdpi` is configured on all required interfaces of your MS-MPC, or of your MX-SPC3 services card if you have enabled Next Gen Services on the MX240, MX480, or MX960. To review how to configure the package on your MS-MPC or MX-SPC3 services card:

- For Junos OS Subscriber Aware, see *Preconfigured Groups for Service PICs and for Session PICs Overview* .
- For Junos OS Broadband Subscriber Management, see "Installing Services Packages for Subscriber Management Application-Aware Policy Management" on page 447.

To configure one or more custom application signatures:

1. Specify a name for the application.

```
[edit services application-identification]
user@host# edit application application-name
```

For example:

```
[edit services application-identification]
user@host# edit application my:http
```

2. Specify a description for the application.

```
[edit services application-identification application application-name]
user@host# set description description
```

For example:

```
[edit services application-identification application my:http]
user@host# set description "Test application"
```

3. Specify an alternative name for the application.

```
[edit services application-identification application application-name]
user@host# set alt-name alt-name
```

For example:

```
[edit services application-identification application my:http]
user@host# set alt-name my:http-app
```

4. Enable saving of the application system cache (ASC).

```
[edit services application-identification application my:http]
user@host# set cacheable
```

5. Specify the name of the Junos OS release for compatibility.

```
[edit services application-identification application application-name]
user@host# set compatibility junos-compatibility-version
```

For example:

```
[edit services application-identification application my:http]
user@host# set compatibility 17.1
```

6. Specify any desired application tags, consisting of a user-defined name and value.

```
[edit services application-identification application application-name]
user@host# set tags tag-name tag-value
```

For example:

```
[edit services application-identification application my:http]
user@host# set tags traffic-type video-stream
```

7. Specify one or more address-based signatures.
   - Specify a destination address and destination port-range.

```
[edit services application-identification application application-name]
user@host# set filter ip 200.0.0.2/24 port-range [80]
```

8. Specify an ICMP-based signature.
   a. Specify ICMP type and code.

```
[edit services application-identification application application-name]
user@host# set icmp-mapping type icmp-type code icmp-code
```

For example:

```
[edit services application-identification application my:http]
user@host# set icmp-mapping type 33 code 34
```

9. Specify an IP protocol-based signature.
   a. Specify the IP protocol by protocol number.

```
[edit services application-identification application application-name]
user@host# set ip-protocol-mapping protocol protocol-number
```

For example:

```
[edit services application-identification application my:http]
user@host# set ip-protocol-mapping protocol 103
```

All ip-protocol-mappings are allowed except Protocol numbers 1,6,17 are not allowed to be configured under ip-protocol based signatures. If you try to configure protocols 1,6,17 under ip-protocol-mapping you will get commit errors.

10. Specify one or more Layer 4 and Layer 7 signatures using pattern matching in conjunction with a Layer 4 protocol.

    a. Specify a name for the Layer 4 and Layer 7 signature.

    ```
    [edit services application-identification application application-name over protocol-type]
    user@host# set signature l4-l7-signature-name
    ```

    For example:

    ```
    [edit services application-identification application my:http over http]
    user@host# set signature myl3l7
    ```

    b. Specify the order to be used if conflicts occur during the application classification. In such a case, the application with lowest order is classified.

    ```
    [edit services application-identification application application-name over protocol-type signature l4-l7-signature-name member member-name]
    user@host# set order order
    ```

    For example:

    ```
    [edit services application-identification application my:http over http signature myl3l7 member m01]
    user@host# set order 1
    ```

c. Specify the priority for using this signature instead of using any matched predefined signatures.

```
[edit services application-identification application application-name over protocol-
type signature l4-l7-signature-name]
user@host# set order-priority (high | low)
```

For example:

```
[edit services application-identification application my:http over http signature myl3l7]
user@host# set order-priority high
```

d. (Optional) Specify the protocol. If you are using Next Gen Services with the MX-SPC3 services card, do not perform this step.

```
[edit services application-identification application application-name over protocol-
type signature l4-l7-signature-name]
user@host# set protocol (http | ssl | tcp | udp)
```

For example:

```
[edit services application-identification application my:http over http signature myl3l7]
user@host# set protocol http
```

e. (Optional) Specify that members are to be matched in order.

```
[edit services application-identification application application-name over protocol-
type signature l4-l7-signature-name]
user@host# set chain-order
```

f. Specify a member. You can repeat this step to define up to four members.

```
[edit services application-identification application application-name over protocol-
type signature l4-l7-signature-name]
user@host# edit member member-name
```

For example:

```
[edit services application-identification application my:http over http signature myl3l7]
user@host# edit member m01
```

g. Specify the member's identifying pattern.

```
[edit services application-identification application application-name over protocol-
type signature l4-l7-signature-name member member-name]
user@host# set pattern pattern
```

For example:

```
[edit services application-identification application my:http over http signature myl3l7
member m01]
user@host# set pattern "www\.facebook\.net"
```

h. Specify the direction of flows to which pattern matching is applied.

```
[edit services application-identification application application-name over protocol-
type signature l4-l7-signature-name member member-name]
user@host# set direction (any | client-to-server | server-to-client)
```

For example:

```
[edit services application-identification application my:http over http signature myl3l7
member m01]
user@host# set direction any
```

i. Specify the number of check-bytes. This option applies to TCP and UDP only.

```
[edit services application-identification application application-name over protocol-
type signature l4-l7-signature-name member member-name]
user@host# set check-bytes max-bytes-to-check
```

For example:

```
[edit services application-identification application my:http over http signature myl3l7
member m01]
user@host# set check-bytes 5000
```

11. (For Next Gen Services with the MX-SPC3 services card only) After you have committed your changes, you can check the status of the custom signature commitment.

```
[edit services application-identification application my:http over http signature myl3l7
member m01]
user@host> show services application-identification commit-status
```

RELATED DOCUMENTATION

Application Identification Overview

Uninstalling a Predefined Junos OS Application Signature Package

NOTE: Starting in Junos OS Release 19.3R2 and 19.4R1, application identification is also supported for Broadband Subscriber Management if you have enabled Next Gen Services on the MX240, MX480 or MX960 router with the MX-SPC3 card.

To uninstall the current application signature package:

- Enter the uninstall command.

```
user@host> request service application-identification uninstall
```

RELATED DOCUMENTATION

Downloading and Installing Predefined Junos OS Application Signature Packages

CHAPTER 32

Configuring Reporting for Application-Aware Data Sessions

IN THIS CHAPTER

- Logging and Reporting Function for Subscribers | 479
- Log Dictionary for Template Types | 487
- Configuring Logging and Reporting for Subscriber Management | 497
- Installing Services Packages for Subscriber Management Logging and Reporting | 498
- Configuring an LRF Profile for Subscribers | 499
- Applying Logging and Reporting Configuration to a Subscriber Management Service Set | 505
- Configuring the Activation of an LRF Rule by a PCC Rule | 506

Logging and Reporting Function for Subscribers

IN THIS SECTION

- Log and Report Control | 480
- Templates | 480
- HTTP Transaction Logging | 485

The logging and reporting function (LRF) enables you to log data for subscriber application-aware policy control sessions and send that data in an IPFIX format to an external log collector using UDP-based transport. These data session logs can include subscriber information, application information, HTTP metadata, data volume, time-of-day information, and source and destination details.

Starting in Junos OS Release 16.1R4 and in Junos OS Release 17.2R1, LRF is available in Junos OS Broadband Subscriber Management. Starting in Junos OS Release 19.3R2, LRF is available in Junos OS

Broadband Subscriber Management if you have enabled Next Gen Services on the MX240, MX480 or MX960 router with the MX-SPC3 card..

The external collector, which is not a Juniper Networks product, can then use this data to perform analytics that provide you with insights about subscriber and application usage, allowing you to create packages and policies that increase revenue.

Log and Report Control

A subscriber's data sessions are logged and sent to collectors based on an LRF profile that you configure and associate with the subscriber.

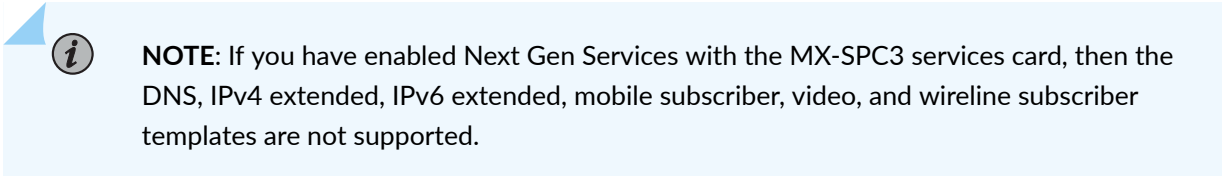
The LRF profile includes:

- Templates—Specify the type of data that you want sent and the trigger that causes data to be sent. You can configure a maximum of 16 templates in an LRF profile.
- Collectors—Identify the destination to send data to. You can configure a maximum of eight collectors in an LRF profile.
- LRF rules—Specify the template and collector to use and, if applicable, a data volume limit that triggers the sending of data. An LRF rule's actions are performed when the matching conditions in a static PCC rule that references the LRF rule are met. You can configure a maximum of 32 LRF rules in an LRF profile.

To associate the LRF profile with a subscriber:

- For Junos OS Subscriber Aware, assign the LRF profile to the subscriber-aware TDF service set that belongs to the TDF interface (mif) in the subscriber's TDF domain.
- For Junos OS Broadband Subscriber Management, assign the LRF profile to the service set that is configured for application-aware policy control.

Templates



NOTE: If you have enabled Next Gen Services with the MX-SPC3 services card, then the DNS, IPv4 extended, IPv6 extended, mobile subscriber, video, and wireline subscriber templates are not supported.

You specify the data fields in a template by configuring one or more types for the template; for example, HTTP and IPv4. Each type represents a set of fields, and the template you configure includes fields from all the types you configure. The template is sent to the collector when you configure it, and is re-sent at a configurable interval. The template types that you can select and the fields that are included by each type are:

- Device Data—Contains data fields specific to the device collecting the logging feed:
  - DPI Engine Version
  - IP address of TDF gateway (in IPv4 format)
- DNS—(Not available if Next Gen Services is enabled with the MX-SPC3 services card) Contains the DNS response time data field.
- Flow ID—Contains the Flow ID data field.

  When HTTP multiple transaction logging is enabled, FlowID is an implicit type that gets included with the HTTP template. When the consolidated session log is generated at the time of SESSION_CLOSE, LRF includes the FlowID that can be used to correlate with the HTTP transaction log records.
- HTTP—Contains data fields for the HTTP metadata from header fields:
  - User Agent
  - Content Length - Request
  - HTTP Response Code
  - Language
  - Host
  - Location
  - Http Method
  - Referer (HTTP)
  - MIME type
  - Time to First Byte
- IFL subscriber— Contains data fields specific to IFL-based subscribers:
  - Subscriber Name—Not applicable for BNG subscribers, hence this value is not be honored (is filled with zero).
  - IFL Name—Filled with default IFL name (filled with values Next Gen Services IFL)
- IPFlow—Contains data fields for the uplink and downlink octets and bytes. When a data record for volume limit is exported, these IPFlow statistics in the record are the actual data received after the last volume limit was reported in that data session and *not* cumulative data.
  - Uplink Octets

  - Downlink Octets
  - Uplink Packets
  - Downlink Packets
  - Ip Protocol—Protocol ID from IP header; for example, 17 (UDP), 6 (TCP).
  - Record Reason—A value of `1` for the session close and a value of `2` for volume-limit.
- IPFlow Extended—Contains data fields for the service set name, routing instance, and payload timestamps. The initiator of the very first packet of a session is the client and the responder is the server.
  - Service-Set-Name—Filled with active `service-set-name` (16 byte value is filled active `service-set-name`. For example, if `service-set-name` is: bng-service-set-1, the template has a value of: bng-service-set-(16bytes)
  - Routing-Instance—Not applicable for BNG subscribers, hence this value is not be honored (is filled with zero).
- IPFlow TCP—Contains data fields for TCP-related timestamps:
  - Retransmitted TCP packets uplink
  - Retransmitted TCP packets downlink
  - TCP flow creation timestamp
- IPFlow TCP Timestamp—Contains IBM-specific data fields for TCP-related timestamps:
  - Smooth RTT uplink
  - Smooth RTT downlink
  - Client setup time
  - Server Setup time
  - First Client Payload timestamp
  - Upload time
  - First Server Payload timestamp
  - Download time
  - Acknowledged volumes uplink
  - Acknowledged volumes downlink

To use the IPFlow TCP Timestamp template when configuring an LRF profile, identify the template as vendor specific to avoid a commit warning. See "Configuring an LRF Profile for Subscribers" on page 499.

- IPFlow Timestamp—Contains data fields for the flow start and end timestamps:
  - Flow Start Time—For TCP, the flow start time is when the SYN packet is received. For UDP, it is when the first packet is sent.
  - Flow End Time
- IPv4—Contains data fields for the basic source and destination IPv4 information:
  - Source IPv4 Address
  - Destination IPv4 Address
- IPv4 Extended—(Not available if Next Gen Services with the MX-SPC3 services card are enabled) Contains data fields for the elements of IPv4 extended fields:
  - IPv4 TOS / Class of Service
  - IPv4 Source Mask
  - IPv4 Destination Mask
  - IPv4 Next Hop
- IPv6—Contains data fields for the basic source and destination IPv6 information:
  - Source IPv6 Address
  - Destination IPv6 Address
- IPv6 Extended—(Not available if Next Gen Services are enabled with the MX-SPC3 services card) Contains data fields for the elements of IPv6 extended fields:
  - IPv6 Source Mask
  - IPv6 Destination Mask
  - IPv6 Next Hop
  - Traffic Class
- L7 Application—Contains data fields for the Layer 7 application:
  - Application Protocol—Application data protocol below the classified application name; for example, `http` or `ssl`.

    - Application Name—Application name; for example, `junos:facebook` or `junos:Netflix`.
    - Host—HTTP header host when application protocol is `http`, SSL common name when application protocol is `ssl`, DNS name when application protocol is `dns`.
- Mobile Subscriber—(Not available if Next Gen Services with the MX-SPC3 services card are enabled) Contains data fields specific to mobile subscribers:
    - IMSI
    - MSISDN
    - IMEI
    - RAT-type
    - ULI
    - RADIUS Called Station ID
- PCC—Contains the PCC rule name data field.Not applicable if Next Gen Services are enabled.
- Status Code Distribution—Contains data fields for the HTTP or DNS status codes:
    - Status code 1
    - Status code 2
    - Status code 3
    - Status code 4
    - Status code 5
    - Num Instances 1
    - Num Instances 2
    - Num Instances 3
    - Num Instances 4
    - Num Instances 5
- Subscriber Data—Contains data fields for Generic Subscriber information that can be included with wireless (mobile) subscribers or wireline subscribers:
    - NAS_IP_ADDR—Not applicable for BNG subscribers, hence this value is not be honored (is filled with zero).
    - Subscriber Type—1 for IP-based subscriber, 2 for IFL-based subscriber.

  - Subscriber IP Address
  - Subscriber VRF—Not applicable for BNG subscribers, hence this value is not be honored (is filled with zero).
  - NAS Port ID—Not applicable for BNG subscribers, hence this value is not be honored (is filled with zero).
  - Accounting-Session-Id—Not applicable for BNG subscribers, hence this value is not be honored (is filled with zero).
  - Class—Not applicable for BNG subscribers, hence this value is not be honored (is filled with zero).
  - NAS Port Type—Not applicable for BNG subscribers, hence this value is not be honored (is filled with zero).
- Transport Layer—Contains data fields for the transport layer:
  - Source Transport Port
  - Destination Transport Port
- Video—(Not available if Next Gen Services with the MX-SPC3 services card are enabled) Contains data fields for video traffic:
  - Bitrate
  - Duration
- Wireline Subscriber—(Not available if Next Gen Services with the MX-SPC3 serices card are enabled) Contains the UserName data field for wireline subscribers. This is the same as RADIUS Called Station ID.

The template that is specified in an LRF rule determines the set of data fields that are included when data is sent to a collector. The data message includes a pointer to the template ID so that the collector can correlate the data contents with the data field lengths and types.

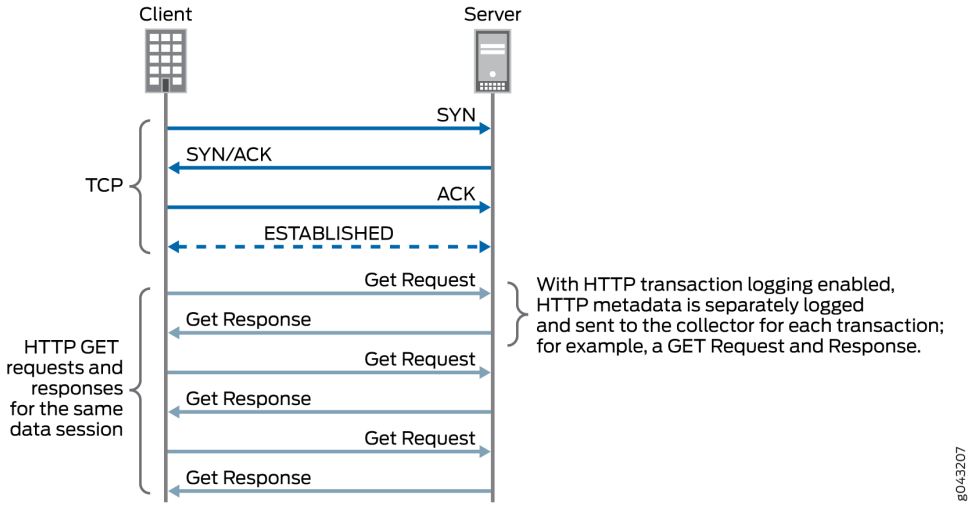
In a template, you also specify the type of trigger that determines when to send data to the collector. This trigger type can be a data volume limit, a time limit, or the closing of a data session (UDP sessions are considered closed after 60 seconds of inactivity; TCP sessions are considered closed when a FIN, FIN-ACK, or RST is received).

HTTP Transaction Logging

You may enable HTTP transaction logging in an LRF profile. This causes each HTTP transaction in a TCP session to be separately logged and sent to the collector, as shown in Figure 7 on page 486. This option is only relevant when the template being used includes HTTP in the template type.

By default, HTTP transaction logging is disabled, and the HTTP transaction records for a TCP session are sent together as one group of records.

Figure 7: HTTP Transaction 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
19.3R1	Starting in Junos OS Release 19.3R2, LRF is available in Junos OS Broadband Subscriber Management if you have enabled Next Gen Services on the MX240, MX480 or MX960 router with the MX-SPC3 card..
16.1R4	Starting in Junos OS Release 16.1R4 and in Junos OS Release 17.2R1, LRF is available in Junos OS Broadband Subscriber Management.

RELATED DOCUMENTATION

Log Dictionary for Template Types | 487

Configuring Logging and Reporting for Junos OS Subscriber Aware

Configuring Logging and Reporting for Subscriber Management | 497

Log Dictionary for Template Types

Table 37 on page 487 shows the logging dictionary of the template types that LRF supports. The log fields are a mix of IETF standard fields and fields that Juniper Networks defined. The IPFIX convention for vendor-defined fields is an enterprise bit set to `1` and an enterprise ID set to the vendor-ID. (The Juniper Networks vendor-ID is 2636.) An IETF standard field has an enterprise bit set to `0` and no value for the enterprise ID.

NOTE: If you have enabled Next Gen Services with the MX-SPC3 services card, then the DNS, IPv4 extended, IPv6 extended, mobile subscriber, video, and wireline subscriber templates are not supported.

Table 37: Logging Dictionary for Template Types

Template Type	Field Name	Enterprise Bit/ID	Information Element Identifier	Data Type	Data Length (bytes)
Device Data	DPI Engine Version	1/2636	503	string	32
	IP address of TDF gateway.	1/2636	502	ipv4Address	4
DNS (Not available if Next Gen Services with the MX-SPC3 services card are enabled)	DNS response time	1/2636	876	dateTimeMilliseconds	8
Flow ID	Flow ID	1/2636	107	unsigned32	4
HTTP	User Agent	1/2636	152	string	32
	Content Length - Request	1/2636	154	unsigned32	4

Table 37: Logging Dictionary for Template Types *(Continued)*

Template Type	Field Name	Enterprise Bit/ID	Information Element Identifier	Data Type	Data Length (bytes)
	HTTP Response Code	1/2636	155	unsigned16	2
	Language	1/2636	156	string	16
	Host	1/2636	157	string	64
	Location	1/2636	158	string	64
	Http Method	1/2636	159	string	8
	Referer(HTTP)	1/2636	160	string	64
	MIME type	1/2636	161	string	32
	Http URI	1/2636	163	string	255
	Time to First Byte	1/2636	181	dateTimeMilliseconds	8
IFL Subscriber	Subscriber Name	1/2636	511	string Not applicable for BNG subscribers, hence this value is not be honored (is filled with zero).	16

Table 37: Logging Dictionary for Template Types *(Continued)*

Template Type	Field Name	Enterprise Bit/ID	Information Element Identifier	Data Type	Data Length (bytes)
	IFL Name	1/2636	512	string Filled with default IFL name (filled with values Next Gen Services IFL)	16
IPFlow	Uplink Octets	1/2636	103	unsigned32	4
	Downlink Octets	1/2636	104	unsigned32	4
	Uplink Packets	1/2636	105	unsigned32	4
	Downlink Packets	1/2636	106	unsigned32	4
	Ip Protocol	0	4	unsigned8	1
	Record Reason	1/2636	112	unsigned8	1

Table 37: Logging Dictionary for Template Types *(Continued)*

Template Type	Field Name	Enterprise Bit/ID	Information Element Identifier	Data Type	Data Length (bytes)
IPFlow Extended	Service-Set-Name	1/2636	520	string Contains data fields for the `service-set-name`, `routing-instance`, and payload timestamps. The initiator of the very first packet of a session is the client and the responder is the server. Filled with active `service-set-name` (16 byte value is filled active `service-set-name`. For example, if `service-set-name` is: bng-service-set-1, the template has a value of: bng-service-set-(16bytes)	16
	Routing-Instance	1/2636	521	string Not applicable for BNG subscribers, hence this value is not be honored (is filled with zero).	16

Table 37: Logging Dictionary for Template Types *(Continued)*

Template Type	Field Name	Enterprise Bit/ID	Information Element Identifier	Data Type	Data Length (bytes)
IPFlow TCP Timestamp	Retransmitted TCP packets uplink	1/2636	115	unsigned32	4
	Retransmitted TCP packets downlink	1/2636	116	unsigned32	4
	Smooth RTT uplink	1/2636	117	dateTimeMilliseconds	8
	Smooth RTT downlink	1/2636	118	dateTimeMilliseconds	8
	Client setup Time	1/2636	119	dateTimeMilliseconds	8
	Server Setup time	1/2636	120	dateTimeMilliseconds	8
	TCP flow creation timestamp	1/2636	121	dateTimeMilliseconds	8
	First Client Payload TS	1/2636	108	dateTimeMilliseconds	8
	Upload time	1/2636	113	dateTimeMilliseconds	8
	First Server Payload TS	1/2636	110	dateTimeMilliseconds	8

Table 37: Logging Dictionary for Template Types *(Continued)*

Template Type	Field Name	Enterprise Bit/ID	Information Element Identifier	Data Type	Data Length (bytes)
	Download time	1/2636	114	dateTimeMilliseconds	8
	Acknowledged volumes uplink	1/2636	122	unsigned64	8
	Acknowledged volumes downlink	1/2636	123	unsigned64	8
IPFlow Timestamp	Flow Start Time	1/2636	101	dateTimeMilliseconds	8
	Flow End Time	1/2636	102	dateTimeMilliseconds	8
IPv4	Source IPv4 Address	0	8	ipv4Address	4
	Destination IPv4 Address	0	12	ipv4Address	4
IPv4 Extended (Not available if Next Gen Services with the MX-SPC3 services card are enabled)	IPv4 TOS/Class of Service	0	5	unsigned8	1
	IPv4 Source Mask	0	9	unsigned8	1
	IPv4 Destination Mask	0	13	unsigned8	1

Table 37: Logging Dictionary for Template Types *(Continued)*

Template Type	Field Name	Enterprise Bit/ID	Information Element Identifier	Data Type	Data Length (bytes)
	IPv4 Next Hop	0	15	ipv4Address	4
IPv6	Source IPv6 Address	0	27	ipv6Address	16
	Destination IPv6 Address	0	28	ipv6Address	16
IPv6 Extended (Not available if Next Gen Services are enabled on the MX-SPC3 services card)	IPv6 Source Mask	0	29	unsigned8	1
	IPv6 Destination Mask	0	30	unsigned8	1
	IPv6 Next hop	0	62	ipv6Address	16
	Traffic Class	1/2636	126	unsigned8	1
L7 Application	Application Protocol	1/2636	151	string	32
	Application Name	1/2636	170	string	32
	Host	1/2636	157	string	64
Mobile Subscriber (Not available if Next Gen Services are enabled on the	IMSI	1/2636	504	string	16
	MSISDN	1/2636	505	string	16

Table 37: Logging Dictionary for Template Types *(Continued)*

Template Type	Field Name	Enterprise Bit/ID	Information Element Identifier	Data Type	Data Length (bytes)
MX-SPC3 services card)	IMEI	1/2636	506	string	16
	RAT-type	1/2636	507	unsigned8	1
	ULI	1/2636	508	string	13
	RADIUS Called Station ID	1/2636	509	string	32
PCC	PCC rule name	1/2636	901	string Not applicable if Next Gen Services are enabled.	64
Status Code Distribution	Status code 1	1/2636	171	unsigned16	2
	Status code 2	1/2636	172	unsigned16	2
	Status code 3	1/2636	173	unsigned16	2
	Status code 4	1/2636	174	unsigned16	2
	Status code 5	1/2636	175	unsigned16	2
	Num Instances 1	1/2636	176	unsigned16	2
	Num Instances 2	1/2636	177	unsigned16	2

Table 37: Logging Dictionary for Template Types *(Continued)*

Template Type	Field Name	Enterprise Bit/ID	Information Element Identifier	Data Type	Data Length (bytes)
	Num Instances 3	1/2636	178	unsigned16	2
	Num Instances 4	1/2636	179	unsigned16	2
	Num Instances 5	1/2636	180	unsigned16	2
Subscriber Data	NAS_IP_ADDR	1/2636	519	ipv4Address Not applicable for BNG subscribers, hence this value is not be honored (is filled with zero).	4
	Subscriber Type	1/2636	515	unsigned8 1 for IP-based subscriber, 2 for IFL-based subscriber	1
	Subscriber IP address	1/2636	516	ipv4Address	4
	Subscriber VRF	1/2636	517	unsigned32 Not applicable for BNG subscribers, hence this value is not be honored (is filled with zero).	4

Table 37: Logging Dictionary for Template Types *(Continued)*

Template Type	Field Name	Enterprise Bit/ID	Information Element Identifier	Data Type	Data Length (bytes)
	NAS Port ID	1/2636	518	string Not applicable for BNG subscribers, hence this value is not be honored (is filled with zero).	32
	Accounting-Session-Id	1/2636	514	string Not applicable for BNG subscribers, hence this value is not be honored (is filled with zero).	32
	Class	1/2636	522	String Not applicable for BNG subscribers, hence this value is not be honored (is filled with zero).	32
	NAS Port Type	1/2636	523	unsigned32 Not applicable for BNG subscribers, hence this value is not be honored (is filled with zero).	4
Transport Layer	Source Transport Port	0	7	unsigned16	2
	Destination Transport Port	0	11	unsigned16	2

Table 37: Logging Dictionary for Template Types *(Continued)*

Template Type	Field Name	Enterprise Bit/ID	Information Element Identifier	Data Type	Data Length (bytes)
Video (Not available if Next Gen Services are enabled on the MX-SPC3 services card)	Bitrate	1/2636	851	unsigned32	2
	Duration	1/2636	852	unsigned32	4
Wireline Subscriber (Not available if Next Gen Services are enabled on the MX-SPC3 services card)	UserName	1/2636	513	string	32

Configuring Logging and Reporting for Subscriber Management

To configure logging and reporting for traffic belonging to a subscriber, you configure LRF rules, collectors, and templates in an LRF profile; assign that LRF profile to the service set that is configured for application-aware policy control, and assign each LRF rule to a PCC rule to activate it.

NOTE: Starting in Junos OS Release 19.3R1, LRF is also supported for Broadband Subscriber Management if Next Gen Services are enabled on the MX-SPC3 services card).

To configure logging and reporting:

1. Install the LRF service package on any MS-MPC PICs that perform logging and reporting or on the MX-SPC3 services card if you have enabled if Next Gen Services.

   See "Installing Services Packages for Subscriber Management Logging and Reporting" on page 498.
2. Configure an LRF profile to specify a set of logging and reporting parameters, which includes data templates, collectors, and LRF rules.

See "Configuring an LRF Profile for Subscribers" on page 499.

3. Assign the LRF profile to the service set that is configured for application-aware policy control.

   See "Applying Logging and Reporting Configuration to a Subscriber Management Service Set" on page 505.

4. Configure activation of an LRF rule with a static PCC rule.

   See "Configuring the Activation of an LRF Rule by a PCC Rule" on page 506.

RELATED DOCUMENTATION

Logging and Reporting Function for Subscribers | 479

Installing Services Packages for Subscriber Management Logging and Reporting

You must install the LRF service package on any MS-MPC PICs that perform logging and reporting or on an MX-SPC3 services card if you have enabled Next Gen Services.

To install the LRF service package:

1. Specify the MS-MPC PIC or MX-SPC3.

```
[edit chassis]
user@host# edit fpc slot-number pic pic-number
```

2. Install the services packages.

```
[edit chassis fpc slot-number pic pic-number adaptive-services service-package extension-
provider ]
user@host# set package jservices-lrf
```

RELATED DOCUMENTATION

Configuring Logging and Reporting for Subscriber Management | 497

Configuring an LRF Profile for Subscribers

IN THIS SECTION

- Configuring the LRF Profile Name | 499
- Configuring Policy-Based Logging | 500
- (Optional) Configuring HTTP Transaction Logging | 500
- Configuring Collectors | 500
- Configuring Templates | 502
- Configuring Logging and Reporting Rules | 503

NOTE: Starting in Junos OS Release 19.3R1, LRF profiles are also supported for Broadband Subscriber Management if Next Gen Services are enabled on the MX-SPC3 services card.

Configure an LRF profile to specify a set of logging and reporting parameters, which includes data templates, collectors, and LRF rules.

To configure an LRF profile:

Configuring the LRF Profile Name

An LRF profile is identified by a name, which you later specify in the service set for the subscribers.

- Configure a name for the LRF profile.

```
[edit services lrf]
user@host# set profile profile-name
```

For example:

```
[edit services lrf]
user@host# set profile lrf_profile1
```

Configuring Policy-Based Logging

Policy-based logging causes the LRF rules to be activated by PCC rules in a static PCEF profile.

- Configure policy-based logging in the LRF profile.

```
[edit services lrf profile profile-name]
user@host# set policy-based-logging
```

For example:

```
[edit services lrf profile  lrf_profile1]
user@host# set policy-based-logging
```

(Optional) Configuring HTTP Transaction Logging

Configure HTTP transaction logging if you want the HTTP metadata generated and sent separately for each transaction of a data session. This option is only relevant if the template specified in an LRF rule includes `http` in the `template-type`.

- Configure HTTP transaction logging in the LRF profile.

```
[edit services lrf profile profile-name]
user@host# set  http-log-multiple-transactions
```

For example:

```
[edit services lrf profile  lrf_profile1]
user@host# set  http-log-multiple-transactions
```

Configuring Collectors

Configure one or more collectors that you want to receive logging and reporting data when an LRF rule is activated. You can configure up to eight collectors for an LRF profile. For each collector:

1. Configure a name for the collector.

```
[edit services lrf profile profile-name]
user@host# set collector collector-name
```

For example:

```
[edit services lrf profile lrf_profile1]
user@host# set collector collector1
```

2. Specify the destination IP address of the collector.

```
[edit services lrf profile profile-name collector collector-name destination]
user@host# set address collector-address
```

For example:

```
[edit services lrf profile  lrf_profile1 collector collector1 destination]
user@host# set address 192.0.2.5
```

3. Specify the destination port of the collector.

```
[edit services lrf profile profile-name collector collector-name destination]
user@host# set port collector-port-number
```

For example:

```
[edit services lrf profile  lrf_profile1 collector collector1 destination]
user@host# set port 4739
```

4. Configure the source address to be used when exporting data to the collector.

```
[edit services lrf profile profile-name collector collector-name]
user@host# set source-address source-address
```

For example:

```
[edit services lrf profile lrf_profile1 collector collector1]
user@host# set source-address 10.1.1.1
```

Configuring Templates

Configure one or more templates, each of which specifies a set of data to be transmitted when an LRF rule is activated. You can configure up to 16 templates for an LRF profile. For each template:

1. Configure a name for the template.

```
[edit services lrf profile profile-name]
user@host# set template template-name
```

For example:

```
[edit services lrf profile lrf_profile1]
user@host# set template template1
```

2. Configure a format for the template. Only the IPFIX format is supported for this release.

```
[edit services lrf profile profile-name template template-name]
user@host# set format ipfix
```

For example:

```
[edit services lrf profile lrf_profile1 template template1]
user@host# set format ipfix
```

3. Configure the template types, which specify the data fields to include. You must configure at least one type, and you can configure multiple types.

```
[edit services lrf profile profile-name template template-name]
user@host# set template-type template-type
```

For example:

```
[edit services lrf profile lrf_profile1 template template1]
user@host# set template-type http ipv4
```

This example results in a template that includes fields from both the HTTP and IPv4 templates.

NOTE: If you have enabled Next Gen Services on the MX-SPC3 services card, then the DNS, IFL subscriber, IPv4 extended, IPv6 extended, mobile subscriber, video, and wireline subscriber templates are not supported.

4. If you used the `ipflow-tcp-ts` template type, identify it as an IBM template to avoid a commit warning.

```
[edit services lrf profile profile-name]
user@host# set vendor-support ibm
```

5. Configure the interval, in seconds, at which you want the template to be retransmitted to the collector. The interval can be from 10 through 600, and the default is 60.

```
[edit services lrf profile profile-name template template-name]
user@host# set template-tx-interval tx-time
```

For example:

```
[edit services lrf profile lrf_profile1 template template1]
user@host# set template-tx-interval 100
```

6. Configure the type of trigger that causes the generation of data records and transmission to the collector. You can specify the trigger type as either the closing of the data session (default) or a data volume limit. The data volume limit value is specified within an LRF rule.

```
[edit services lrf profile profile-name template template-name]
user@host# set trigger-type (session-close | volume)
```

For example:

```
[edit services lrf profile lrf_profile1 template template1]
user@host# set trigger-type volume
```

Configuring Logging and Reporting Rules

Configure one or more LRF rules, which control how data sessions are logged and reported. You can configure up to 32 LRF rules for an LRF profile. For each LRF rule:

1. Configure a name for the LRF rule.

```
[edit services lrf profile profile-name]
user@host# set rule lrf-rule-name
```

For example:

```
[edit services lrf profile lrf_profile1]
user@host# set rule rule1
```

You cannot use the same LRF rule name in multiple LRF profiles.

2. Specify the collector that you want to receive the data if this rule is matched.

```
[edit services lrf profile profile-name rule lrf-rule-name ]
user@host# set then report collector collector-name
```

For example:

```
[edit services lrf profile lrf_profile1 rule rule1]
user@host# set then report collector collector1
```

3. Specify the template that identifies the type of data to report if this rule is matched.

```
[edit services lrf profile profile-name rule lrf-rule-name]
user@host# set then report template template-name
```

For example:

```
[edit services lrf profile lrf_profile1 rule rule1]
user@host# set then report template template1
```

4. If you specified `volume` for the template's trigger type in Step 6 of "Configuring Templates" on page 502, configure the data volume limit to be used for reporting by this rule.

```
[edit services lrf profile profile-name rule lrf-rule-name]
user@host# set then report volume-limit volume
```

The data volume, in megabytes, can be from 1 through 1024.

For example:

```
[edit services lrf profile lrf_profile1 rule rule1]
user@host# set then report volume-limit 4
```

5. If you specified `time` for the template's trigger type in Step 6 of "Configuring Templates" on page 502, configure the time limit to be used for reporting by this rule.

```
[edit services lrf profile profile-name rule lrf-rule-name]
user@host# set then report time-limit time-interval
```

The time limit, in seconds, can be from 60 through 1800. The default is 300.

For example:

```
[edit services lrf profile lrf_profile1 rule rule1]
user@host# set then report time-limit 360
```

RELATED DOCUMENTATION

Logging and Reporting Function for Subscribers

Applying Logging and Reporting Configuration to a Subscriber Management Service Set | **505**

Configuring the Activation of an LRF Rule by a PCC Rule

Configuring Custom Application Signatures

Applying Logging and Reporting Configuration to a Subscriber Management Service Set

NOTE: Starting in Junos OS Release 19.3R1, LRF profiles are also supported for Broadband Subscriber Management if Next Gen Services are enabled on the MX-SPC3 services card.

To use an LRF profile, you must assign it to the service set that is configured for application-aware policy control.

To assign an LRF profile to subscribers:

- Assign the LRF profile to the service set.

```
[edit services service-set service-set-name]
user@host# set lrf-profile profile-name
```

RELATED DOCUMENTATION

Logging and Reporting Function for Subscribers | 479

Configuring an LRF Profile for Subscribers | 499

Applying Services to Subscriber-Aware Traffic with a Service Set

Configuring Logging and Reporting for Subscriber Management | 497

Identifying the Service Interface That Handles Subscriber Management Application-Aware Policy Control | 462

Configuring the Activation of an LRF Rule by a PCC Rule

NOTE: Starting in Junos OS Release 19.3R1, LRF rules are also supported for Broadband Subscriber Management if Next Gen Services are enabled on the MX-SPC3 services card.

NOTE: If you are using Junos OS Subscriber Aware, you must be in maintenance mode to make a change to a PCC action profile. (See *Changing PCEF Profiles, PCC Rules, PCC Rulebases, Diameter Profiles, Flow Descriptions, and PCC Action Profiles*).

NOTE: If you are using Junos OS Broadband Subscriber Management, you cannot make a change to a PCC action profile that is being used by subscribers. To modify the PCC action profile, you must first log off the subscribers that are using the PCC action profile.

Before you configure activation of an LRF rule by a PCC rule, you must:

- Configure the LRF rule in an LRF profile.
- Configure policy-based logging in the LRF profile.
- Configure the PCC rule.

You use a PCC rule's matching conditions to activate an LRF rule, which controls how data sessions are logged and reported. You identify the LRF rule in the PCC rule's action profile.

You can configure a PCC rule to activate an LRF rule for Junos OS Subscriber Aware or for Junos OS Broadband Subscriber Management, but you use a different CLI hierarchy level for each product.

- If you are using Junos OS Subscriber Aware, configure PCC rules at the `[edit unified-edge pcef]` hierarchy level.
- If you are using Junos OS Broadband Subscriber Management, configure PCC rules at the `[edit services pcef]` hierarchy level.

To configure a PCC rule to activate an LRF rule:

1. Identify the PCC action profile that is used in the PCC rule.

   For Junos OS Subscriber Aware:

   ```
   [edit unified-edge pcef]
   user@host# show pcc-rules rule-name
   ```

   For Junos OS Broadband Subscriber Management:

   ```
   [edit services pcef]
   user@host# show pcc-rules rule-name
   ```

   For example:

   For Junos OS Subscriber Aware:

   ```
   [edit unified-edge pcef]
   user@host# show pcc-rules all-traffic
   ```

   ```
   from {
       flows {
           all;
       }
   }
   then {
       pcc-action-profile all-traffic-action;
   }
   ```

For Junos OS Broadband Subscriber Management:

> **NOTE**: The `from` statement is not applicable for Next Gen Services MX-SPC3 services card.

```
[edit services pcef]
user@host# show pcc-rules all-traffic
```

```
from {
    flows {
        all;
    }
}
then {
    pcc-action-profile all-traffic-action;
}
```

2. Assign the LRF rule to the PCC action profile.

   For Junos OS Subscriber Aware:

```
[edit unified-edge pcef pcc-action-profiles profile-name]
user@host# set logging-rule lrf-rule-name
```

   For Junos OS Broadband Subscriber Management:

```
[edit services pcef pcc-action-profiles profile-name]
user@host# set logging-rule lrf-rule-name
```

   For example:

   For Junos OS Subscriber Aware:

```
[edit unified-edge pcef pcc-action-profiles all-traffic-action]
user@host# set logging-rule rule1
```

For Junos OS Broadband Subscriber Management:

```
[edit services pcef pcc-action-profiles all-traffic-action]
user@host# set logging-rule rule1
```

RELATED DOCUMENTATION

- *Logging and Reporting Function for Subscribers*
- *Configuring an LRF Profile for Subscribers*
- *Configuring Policy and Charging Control Rules*

PART 6

Configuring HTTP Redirect Services

- Configuring Captive Portal Content Delivery Services for Redirected Subscribers | **511**

CHAPTER 33

Configuring Captive Portal Content Delivery Services for Redirected Subscribers

IN THIS CHAPTER

- HTTP Redirect Service Overview | **511**
- Remote HTTP Redirect Server Operation Flow | **518**
- Local HTTP Redirect Server Operation Flow (MX Series, ACX7100-48L, ACX7332 and ACX7348) | **520**
- Configuring MS-MPC-Based or MX-SPC3-Based Static HTTP Redirect Services | **522**
- Configuring MS-MPC-Based or MX-SPC3-Based Converged HTTP Redirect Services | **532**
- Configuring Routing Engine-Based, Static HTTP Redirect Services | **544**
- Configuring Routing Engine-Based, Converged HTTP Redirect Services | **559**
- Adding Subscriber Information to HTTP Redirect URLs | **570**
- How to Automatically Remove the HTTP Redirect Service After the Initial Redirect | **572**
- Example: Configuring HTTP Redirect Services Using a Next-Hop Method and Attaching It to a Static Interface | **575**

HTTP Redirect Service Overview

IN THIS SECTION

- Services-Card-Based Captive Portal | **514**
- Routing Engine-Based Captive Portal | **515**
- Converged Service Provisioning for HTTP Redirect Services | **516**
- Static Service Provisioning for HTTP Redirect Services | **516**

HTTP request traffic from subscribers is aggregated from access networks onto a Broadband Remote Access Server (B-RAS) router, where HTTP traffic can be intercepted and redirected to a captive portal on an external device. The captive portal is often the initial page a subscriber sees after logging in to a subscriber session. The captive portal also receives and manages HTTP requests to unauthorized Web resources.

For example, the user might be redirected to a webpage that shows a company logo and network usage policy or to a page where the subscriber pays for services. The captive portal typically provides authentication and authorization services for redirected subscribers before granting access to protected servers outside of a walled garden.

A *walled garden*, also known as an *allowlist*, defines a group of servers where access is provided to subscribers without reauthorization through a captive portal. These walled gardens enable you to increase revenue by marketing various services to your customers.

Typical walled garden links are:

- Vendor services, such as automobile rentals
- Hotel and motel loyalty or corporate program portals
- Room services
- Local attractions and weather

NOTE: This documentation uses the terms *HTTP redirect service* and *captive portal content delivery (CPCD) service* interchangeably.

The HTTP redirect service implements a data handler and a control handler and registers them with service rules applicable to the HTTP applications. These rules are parsed by the cpcdd process on the Routing Engine. The data handler applies the rules to HTTP data flows and handles rewriting the IP destination address or sending an HTTP response with a preconfigured redirect URL. The response message includes an HTTP status code. The status code that is returned depends on the HTTP version used by the HTTP client that sent the GET request. When the version is higher than HTTP 1.0, the redirect server returns the 307 (Temporary Redirect) status code. When the version is HTTP 1.0, the 302 (Found) status code is returned. In releases earlier than 17.3R1, the redirect server returns the 302 status code regardless of HTTP version. Both codes inform the HTTP client to use the original URL, rather than the redirect URL, for subsequent GET requests.

When the response to the HTTP request is sent to the subscriber, the original URL is preserved by optionally appending it to the end of the configured redirect URL. The maximum length of the redirect URL, including the appended original URL, is 128 bytes. The maximum length of the redirect URL is increased to 1360 bytes and the redirect server can append additional information about the subscriber to the redirect URL. The maximum length applies regardless of whether subscriber information is appended to the URL. To append the subscriber information, you can specify certain subscriber

attributes in the VSAs returned in the RADIUS Accept-Access message in response to the subscriber login or in a RADIUS Change of Authorization (CoA) message. This applies for both Activate-Service (26-65) and Deactivate-Service (26-66) VSAs. The subscriber information is retrieved from the subscriber session database.

The control handler maintains a connection with the cpcdd process on the Routing Engine to learn configuration changes, such as the redirect URL and the rewrite IP destination and port. To achieve faster performance, the control handler maintains a cache of relevant configured entities, such as URLs, on a Modular Port Concentrator (MPC).

HTTP redirect services are supported for both IPv4 and IPv6. You can attach an HTTP redirect service or service set to either a static or dynamic interface. For dynamic subscriber management, you can attach HTTP services or service sets dynamically at subscriber login or by using a RADIUS change of authorization (CoA).

NOTE: On Junos OS Evolved:

- You can configure up to 1000 HTTP redirect services.
- You can create counters for each term in an HTTP Redirect service filter. Because the service filter is installed as a shared filter, only a single aggregate counter is available in hardware for each term as configured. The captive portal content delivery (CPCD) application counts redirected packets in software using an aggregate counter. Use the `show services captive-portal-content-delivery statistics interface` command to view the counter statistics.

There are three methods to configure HTTP redirect services. You can also configure HTTP redirect on the MX-SPC3 services processing card if Next Gen Services are enabled. Table 38 on page 513 lists the methods supported for HTTP redirect services and the Junos OS releases that support each method.

Table 38: Supported HTTP Redirect Methods by Release

Method		Junos OS Releases Supported
MS-DPC-based		(Not supported for Next Gen Services on the MX-SPC3 services card)
	Static	Releases earlier than 15.1
	Converged	Not supported

Table 38: Supported HTTP Redirect Methods by Release *(Continued)*

Method		Junos OS Releases Supported
MS-MPC-based		(Not supported for Next Gen Services on the MX-SPC3 services card.)
	Static	Starting in Junos OS Release 15.1
	Converged	Starting in Junos OS Release 17.2
MX-SPC3-based		
	Static	Starting in Junos OS Release 19.3R2 if Next Gen Services are enabled on the MX-SPC3 services card.
	Converged	Starting in Junos OS Release 19.3R2 if Next Gen Services are enabled on the MX-SPC3 services card.
Routing Engine-based		
	Static	All Junos OS releases
	Converged	Starting in Junos OS Release 16.1R4 and 17.2

For all methods, you configure the walled garden as a static firewall service filter.

Services-Card-Based Captive Portal

MS-MPC–Based Captive Portal

Starting in Junos OS Release 15.1R4, the only line card and interface card combination that supports HTTP redirect services on MX Series routers is the Multiservices Modular Port Concentrator (MS-MPC) with a Multiservices Modular Interface Card (MS-MIC). This combination provides improved scaling and high performance. MS-MICs and MS-MPCs have enhanced memory (16 GB for MS-MIC, 32 GB per NPU of MS-MPC) and processing capabilities. The services interfaces on MS-MPCs and MS-MICs are identified in the configuration with an ms- prefix (for example, ms-1/2/1).

NOTE: Throughout this documentation, the term *MS-MPC–based* refers to MPCs with MS-MICs installed and to MS-MICs alone when they are installed in MX Series routers that do not accept line cards.

MX-SPC3 Services Card-Based Captive Portal

Starting in Junos OS Release 19.3R2, you can configure HTTP redirect services if Next Gen Services are enabled on the MX-SPC3 services card. The services interfaces on MX-SPC3s are identified in the configuration with a vms- prefix (for example, vms-1/2/1).

Walled Garden Configured as a Service Filter

Packet flow for a services-card-based captive portal differs depending on how you configure the walled garden. HTTP traffic destined to servers within the walled garden does not flow to the services card. However, any HTTP traffic destined outside of the walled garden flows to the services card.

- For subscriber requests contained within the first packet of data traffic, the system expects TCP proxy to generate a TCP SYN flag causing the data handler to perform a rule lookup and apply those rules to HTTP data flows.
  - For an HTTP rewrite condition—If the IP destination address is not provided in the policy, the control handler looks up the IP destination address.
  - For an HTTP redirect condition—TCP proxy is triggered to complete its three-way handshake.
- For HTTP request packets.
  - For an HTTP rewrite condition—The control handler uses the cached IP destination address and modifies the data packet.
  - For an HTTP redirect condition—The control handler sends an HTTP 302 or 307 response with a preconfigured redirect URL.

Routing Engine-Based Captive Portal

The Routing Engine-based captive portal supports a walled garden as a firewall service filter for both static and converged services. As soon as the HTTP traffic matches the rules defined in the firewall service filter, the HTTP traffic is sent to the Routing Engine. The services interfaces on the Routing Engine are identified with an si- prefix (for example, si-1/1/0). The si- interface handles all redirect and rewrite traffic and services for the Routing Engine. The si- interface must be operational with a status of up to enable and activate the captive portal content delivery (CPCD) service. After the CPCD service is enabled, any change in the operational state of the si- interface does not affect existing CPCD services.

Converged Service Provisioning for HTTP Redirect Services

Starting in Junos OS Release 17.2R1, converged service provisioning is supported for both Routing Engine-Based and MS-MPC/MS-MIC–based captive portals. Starting in Junos OS Release 19.3R2, converged service provisioning is also supported for MX-SPC3 services card–based captive portals if Next Gen Services are enabled on the MX-SPC3 services card. Converged service provisioning means you can configure service provisioning in a dynamic profile. You can specify user-defined variables for services that are populated by means of a RADIUS VSA or a Change of Authorization (CoA) message.

For example, you might want to have a different redirect URL for each subscriber. You can create a redirect-url variable in the dynamic profile, then configure a service rule to redirect the matching subscriber to $redirect-url. When RADIUS authenticates the user, the Activate-Service VSA (26–65) provides the URL specific to that user.

Static Service Provisioning for HTTP Redirect Services

Starting in Junos OS Release 17.4R1, static service provisioning is supported for both Routing Engine-Based and MS-MPC/MS-MIC–based captive portals. Starting in Junos OS Release 19.3R2, static service provisioning is also supported for MX-SPC3–based captive portals if Next Gen Services are enabled on the MX-SPC3 services card. Static service provisioning means you can configure service provisioning in a static profile. You can specify user-defined variables (for example, `http://portal.wifi.example.com/xx?wlanuseraddr=%subsc-ip%&nasaddr=%nas-ip%&acname=%ac -name%&url=%dest-url%&userlocation=%nas-port-id%&usermac=%mac-sa%& session-id=%sess-id%&username=%user-name%&wlanuseraddrv6=%subsc-ipv6%`) for services that are populated by means of a RADIUS VSA or a Change of Authorization (CoA) message.

In static CPCD, attributes in a redirect URL are not sent in the Juniper Networks VSAs, Activate-Service (26-65) and Deactivate-Service (26-66). You can configure it as shown in the following example:

```
captive-portal-content-delivery {
    rule redirect {
        match-direction input;
        term t1 {
            then {
                redirect url;
            }
        }
    }
}
```

The tokens in the url such as “subsc-ip”, “nas-ip”, “ac-name” must be specified between “%” symbol. The order of tokens does not matter.

Following is a list of token with their significance:

- %subsc-ip%—private IP address of the subscriber.
- %nas-ip%—BNG IP address.
- %ac-name%—It will be empty for the BNG.
- %dest-url%—The original request url.
- %nas-port-id%—Used for subscriber. This parameter must include interface name, pvlan and cvlan. The interface name could be physical or virtual interface name. For example, ge0/0/0 or ae0. The pvlan and cvlan range is 14095
- %mac-sa%—WLAN client MAC address.
- %sess-id%—session-id of subscriber.
- %user-name%—username of a subscriber.
- %subsc-ipv6%—subscriber IPv6 address (only IANA address). If IANA address is not specified for the subscriber, this field will be empty.

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
25.4R1	Starting in Junos OS Evolved Release 25.4R1, you can enable HTTP redirect service on Junos OS Evolved Broadband Network Gateways (BNGs).
19.3R2	Starting in Junos OS Release 19.3R2, HTTP redirect can also be configured on the MX-SPC3 services processing card if Next Gen Services are enabled.
19.3R2	Starting in Junos OS Release 19.3R2, you can configure HTTP redirect services if Next Gen Services are enabled on the MX-SPC3 services card.
19.3R2	Starting in Junos OS Release 19.3R2, converged service provisioning is also supported for MX-SPC3 services card–based captive portals if Next Gen Services are enabled on the MX-SPC3 services card.
19.3R2	Starting in Junos OS Release 19.3R2, static service provisioning is also supported for MX-SPC3–based captive portals if Next Gen Services are enabled on the MX-SPC3 services card.

RELATED DOCUMENTATION

Local HTTP Redirect Server Operation Flow (MX Series, ACX7100-48L, ACX7332 and ACX7348) | 520

Remote HTTP Redirect Server Operation Flow | 518

Configuring MS-MPC-Based or MX-SPC3-Based Static HTTP Redirect Services | 522

Configuring MS-MPC-Based or MX-SPC3-Based Converged HTTP Redirect Services | 532

Configuring Routing Engine-Based, Static HTTP Redirect Services | 544

Configuring Routing Engine-Based, Converged HTTP Redirect Services | 559

Adding Subscriber Information to HTTP Redirect URLs | 570

How to Automatically Remove the HTTP Redirect Service After the Initial Redirect | 572

Remote HTTP Redirect Server Operation Flow

You can use the remote HTTP redirect feature in configurations where the redirect server resides outside of the MX Series router and on a policy server, such as Session and Resource Control (SRC).

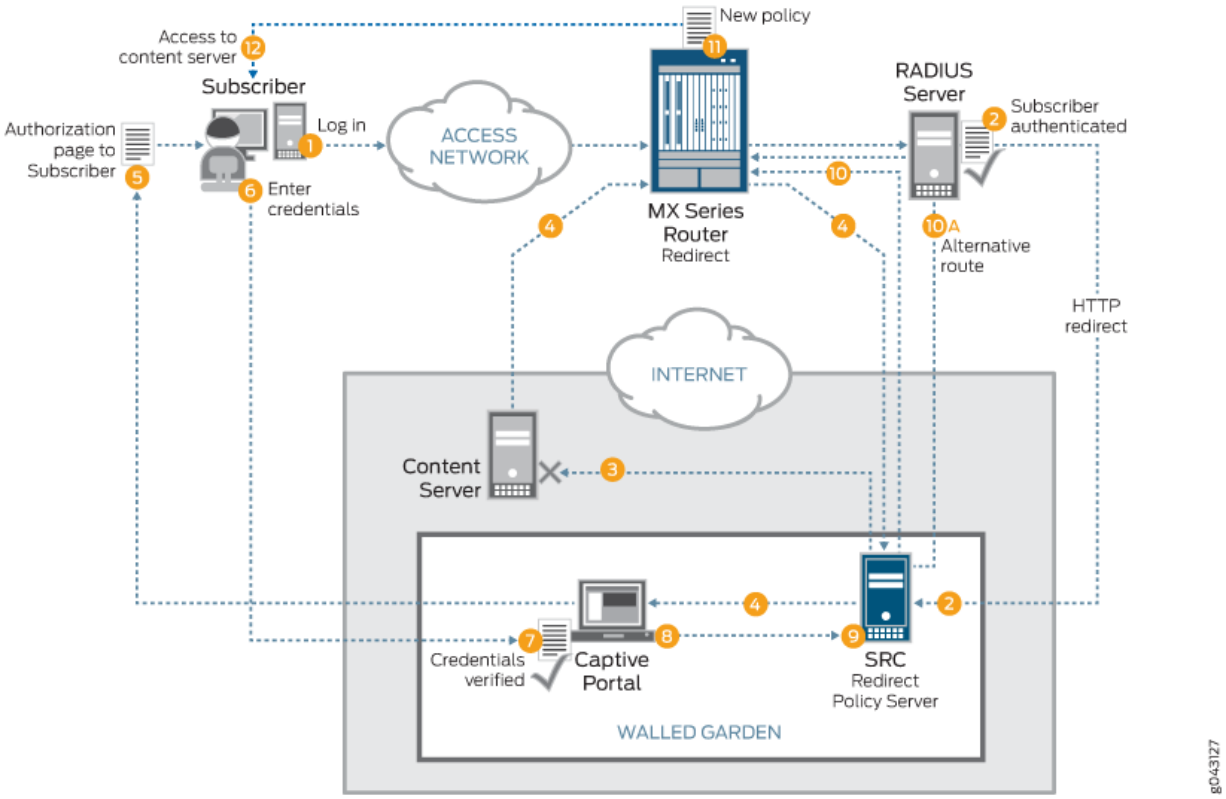
An HTTP redirect remote server that resides in a walled garden behind the router processes HTTP requests redirected to it and responds with a redirect URL that points to a captive portal. When you use a remote HTTP redirect server, you need to configure an HTTP service rule that rewrites the IP destination address of the incoming HTTP requests on the service router. The rewritten address ensures that the requests reach the remote HTTP redirect server before being redirected to a captive portal.

HTTP traffic is intercepted at the broadband network gateway (BNG) and the IP destination address is rewritten so that the HTTP requests are sent to the HTTP redirect server instead of the original destination. The HTTP redirect server sends a response with the HTTP 302 or 307 status code with the URL of the designated captive portal using either IPv4 destination address/destination port rewrite, or IPv6 destination address/destination port rewrite.

Figure 8 on page 519 shows the general service deployment during access configuration for a remote HTTP redirect server. The HTTP redirect server resides outside of the MX Series router on a policy server, such as SRC. Service attachment occurs at subscriber login, and service detachment occurs at subscriber logout.

NOTE: A complete HTTP redirect solution depends on back-end servers, such as SRC, captive portal, and RADIUS, and their integration specific to each customer's favored integration scheme.

Figure 8: Remote HTTP Redirect Server Deployment



1. The subscriber logs in connecting through the access network.
2. RADIUS authenticates the subscriber and sends a service activate (IP destination address rewrite), which redirects HTTP traffic to the redirect policy server (such as SRC) in a walled garden.
3. The subscriber attempts to access the content server (HTTP traffic).
4. The subscriber's HTTP traffic is first redirected to the SRC redirect policy server, which then redirects it to the captive portal.
5. The captive portal sends an authorization page back to the subscriber.
6. The subscriber enters credentials to obtain authorization.
7. The captive portal verifies the subscriber's credentials.
8. The captive portal authorizes the subscriber and notifies the SRC redirect policy server.
9. The SRC redirect policy server checks the subscriber database and formulates a policy so the subscriber can access the content server.

10. The SRC redirect policy server sends the policy directly to the MX Series router using JSRC or Diameter.

    Alternatively, the SRC redirect policy server notifies the RADIUS server, which in turn sends a change of authorization (CoA) to the MX Series router.

11. The MX Series router attaches the new policy, overriding the initial redirect policy.

12. The subscriber now gains access to the content server.

RELATED DOCUMENTATION

HTTP Redirect Service Overview | 511

Local HTTP Redirect Server Operation Flow (MX Series, ACX7100-48L, ACX7332 and ACX7348) | 520

Local HTTP Redirect Server Operation Flow (MX Series, ACX7100-48L, ACX7332 and ACX7348)

You can use the local HTTP redirect feature in configurations where the redirect server resides locally on a router. This feature is available on MX series routers and ACX7100-48L, ACX7332 and ACX7348 routers.

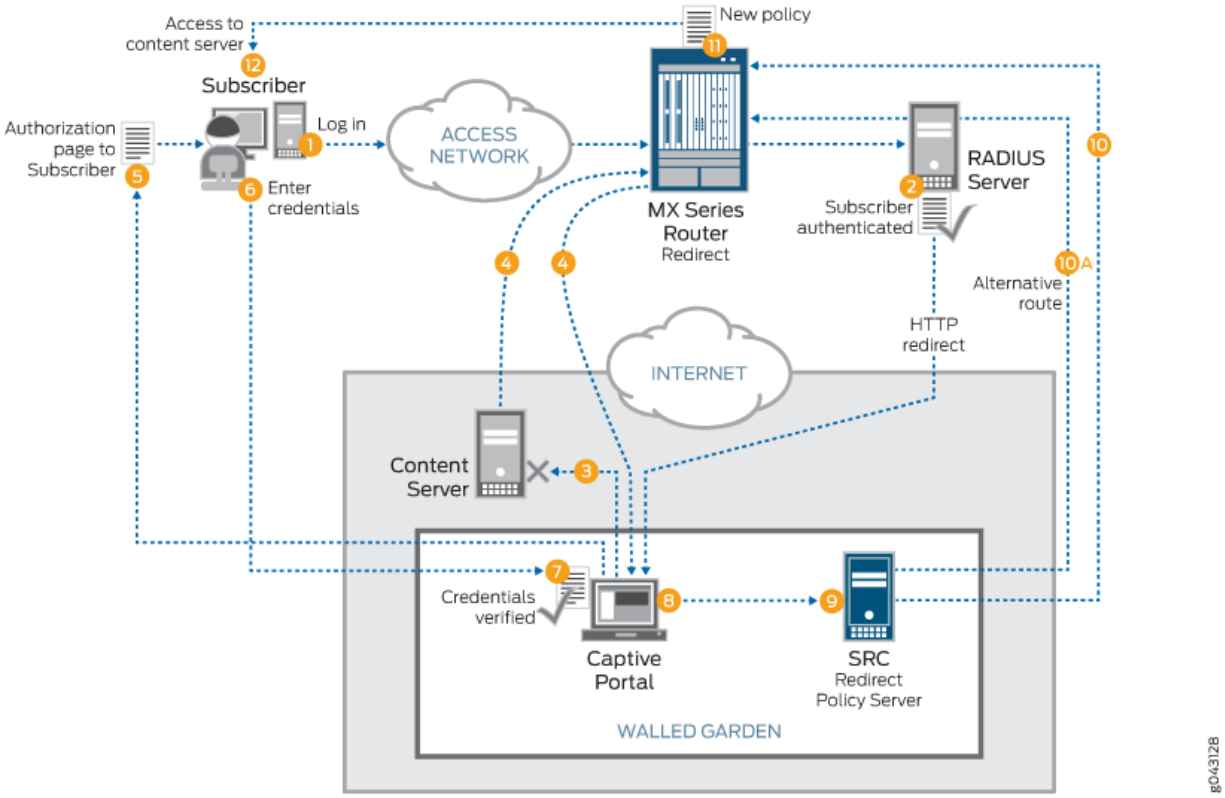
An HTTP redirect local server that resides locally on a router processes HTTP requests redirected to it and responds with a redirect URL that points to a captive portal. You can implement the local server as a service within a service set, which provides more scalability and better performance. When you use a local HTTP redirect server, you need to configure an HTTP service rule to redirect HTTP requests to a captive portal within a walled garden.

A walled garden is a group of servers that provide subscriber access to sites within the walled garden without requiring reauthorization through a captive portal. HTTP request traffic from subscribers destined to servers outside of the walled garden is intercepted and redirected by either the captive portal content delivery (CPCD) service or the Routing Engine. The CPCD service or Routing Engine locates the provisioned redirect URL for the specific subscriber and sends a response with the HTTP 302 or 307 status code that includes the located URL.

Figure 9 on page 521 shows the general service deployment during access configuration for a local HTTP redirect server. The HTTP redirect server resides locally on the router. Service attachment occurs at subscriber login, and service detachment occurs at subscriber logout.

NOTE: A complete HTTP redirect solution depends on back-end servers, such as SRC, captive portal, and RADIUS; their integration is specific to each customer's favored integration scheme.

Figure 9: Local HTTP Redirect Server Deployment



1. The subscriber logs in connecting through the access network.
2. RADIUS authenticates the subscriber and sends a service activate (HTTP redirect), which redirects HTTP traffic to the captive portal in a walled garden.
3. The subscriber attempts to access the content server (HTTP traffic) outside the walled garden.
4. The subscriber's HTTP traffic is redirected to the captive portal by the router.
5. The captive portal sends an authorization page back to the subscriber.
6. The subscriber enters credentials to obtain authorization.

7. The captive portal verifies the subscriber credentials.
8. The captive portal authorizes the subscriber and notifies the SRC redirect policy server.
9. The SRC redirect policy server checks the subscriber database and formulates a policy so the subscriber can access the content server.
10. The SRC redirect policy server sends the policy directly to the router using JSRC or Diameter.

    Alternatively, the SRC redirect policy server notifies the RADIUS server, which in turn sends a change of authorization (CoA) to the router.
11. The router attaches the new policy, overriding the initial redirect policy.
12. The subscriber now gains access to the content server.

RELATED DOCUMENTATION

HTTP Redirect Service Overview | 511

Remote HTTP Redirect Server Operation Flow | 518

Configuring MS-MPC-Based or MX-SPC3-Based Static HTTP Redirect Services

IN THIS SECTION

- Configuring a Walled Garden as a Firewall Service Filter | 523
- Configuring HTTP Redirect for Local and Remote Redirect Servers | 527
- Configuring the Service Profile and the Service Set to Associate the Service Profile with a Service Interface | 528
- Attaching a CPCD Service Set and Service Filter to a Logical Interface | 530
- Installing a Service Package for CPCD Service | 531

NOTE: Starting in Junos OS Release 19.3R2, static HTTP redirect service provisioning is also supported for MX-SPC3 services card–based captive portals if you have enabled Next Gen Services on the MX Series router.

A walled garden is a group of servers that provide subscriber access to sites within the walled garden without requiring reauthorization through a captive portal. The captive portal page is typically the initial page a subscriber sees after logging in to a subscriber session.

When subscribers try to access sites outside the walled garden, HTTP redirect services process IPv4 and IPv6 HTTP requests to manage that traffic. The subscriber HTTP request traffic that is not destined for the walled garden is sent to the redirect server, which responds with a redirect URL that sends traffic to a captive portal instead of to the unauthorized external site. The captive portal provides authentication and authorization services for the redirected subscribers before granting them access to protected servers outside of the walled garden.

The redirect server can be local or remote:

- Local redirect server—Resides on the router and redirects subscriber traffic to a captive portal inside a walled garden.
- Remote redirect server—Resides on a device such as a policy server inside a walled garden behind the router. The destination address for the subscriber's HTTP traffic is rewritten to the address of the remote redirect server. The remote server redirects subscriber traffic to a captive portal inside that walled garden.

You configure the walled garden as a firewall service filter. The service filter is attached to a static interface. The CPCD service is applied to a service interface (ms- on the MS-MPC or vms- on the MX-SPC3 services card) by means of a service set; the service set is then attached to a static interface.

Configuring a Walled Garden as a Firewall Service Filter

When you configure the walled garden as a firewall service filter, traffic that is destined to servers within the walled garden is identified and skipped. Because this traffic does not flow to the line card, handling requirements are reduced.

All other HTTP traffic is destined for addresses outside the walled garden. Because this traffic does not match the filter conditions, it flows to the line card for handling.

You can configure the service filter so that the walled garden contains a single server as the captive portal or a list of servers.

- Configure the walled garden with a single server as the captive portal:

1. Create the service filter.

```
[edit]
user@host# edit firewall family address-family service-filter filter-name
```

2. Define a filter term to identify and skip processing for traffic to the captive portal.

   a. Specify filter conditions to match traffic that is destined for the captive portal by specifying the destination address of the captive portal and the destination port.

```
[edit firewall family inet service-filter filter-name]
user@host# set term name from destination-address ip-address
user@host# set term name from destination-port port-number
```

   b. Specify that the matching traffic skips processing on the line card.

```
[edit firewall family inet service-filter filter-name]
user@host# set term name then skip
```

3. Define a filter term to identify HTTP traffic from all the traffic that did not match the previous term and send it for processing by CPCD service rules.

   a. Specify one or more HTTP port numbers to match the skipped HTTP traffic.

```
[edit firewall family inet service-filter filter-name]
user@host# set term name from destination-port http-port-number
```

   b. Specify that the matching traffic is processed by a CPCD service.

```
[edit firewall family inet service-filter filter-name]
user@host# set term name then service
```

4. Define a filter term to skip further action for any remaining, non-HTTP traffic.

```
[edit firewall family inet service-filter filter-name]
user@host# set term name then skip
```

For example, the following configuration creates a filter for IPv4 HTTP traffic, walled-v4, with the captive portal on 192.0.2.0. Traffic matching the address is skipped. Nonmatching traffic goes to term http, where HTTP traffic is picked out of all skipped traffic and sent to be processed according to a CPCD service. Finally, term skip causes all the remaining, non-HTTP traffic to be skipped.

```
[edit]
user@host# edit firewall family inet service-filter walled-v4
[edit firewall family inet service-filter walled-v4]
user@host# set term portal from destination-address 192.0.2.0
user@host# set term portal from destination-port 80
user@host# set term portal then skip
user@host# set term http from destination-port 80
user@host# set term http then service
user@host# set term skip then skip
```

- Configure the walled garden as a list or subnet of servers.

  1. Create the service filter.

     ```
     [edit]
     user@host# edit firewall family address-family service-filter filter-name
     ```

  2. Define a filter term.

     a. Specify filter conditions to match traffic that is destined for any server in the walled garden by specifying a destination prefix list of servers.

     ```
     [edit firewall family inet service-filter filter-name]
     user@host# set term name from destination-prefix-list list-name
     user@host# set term name from destination-port port-number
     ```

     b. Specify that the matching traffic skips processing on the line card.

     ```
     [edit firewall family inet service-filter filter-name]
     user@host# set term name then skip
     ```

  3. Define a filter term to identify HTTP traffic from all the traffic that did not match the previous term and send it for processing by CPCD service rules.

a. Specify one or more HTTP port numbers to match the skipped HTTP traffic.

```
[edit firewall family inet service-filter filter-name]
user@host# set term name from destination-port http-port-number
```

b. Specify that the matching traffic is processed by a CPCD service.

```
[edit firewall family inet service-filter filter-name]
user@host# set term name then service
```

4. Define a filter term to skip further action for any remaining, non-HTTP traffic.

```
[edit firewall family inet service-filter filter-name]
user@host# set term name then skip
```

5. (Optional) Define a prefix list that specifies servers within the walled garden. You can specify a subnet or multiple individual addresses.

```
[edit policy-options]
user@host# set prefix-list list- name ip-address/mask
user@host# set prefix-list list- name ip-address1
user@host# set prefix-list list- name ip-address2
```

For example, the following configuration creates a filter for IPv6 HTTP traffic, walled-v6-list, with a prefix list, wg-list, that specifies two servers in the walled garden. Filter term portal6 identifies IPv6 traffic that is destined for the walled garden. Nonmatching traffic goes to term http6, where HTTP traffic is picked out of all skipped traffic and sent to be processed according to a CPCD service. Finally, term skip causes all the remaining, non-HTTP traffic to skipped.

```
[edit]
user@host# edit firewall family inet6 service-filter walled-v6-list
user@host# set term portal6 from destination-prefix-list wg-list
user@host# set term portal6 then skip
user@host# set term http6 from destination-port [80 8080]
user@host# set term http6 then service
user@host# set term skip6 then skip
[edit policy-options]
```

```
user@host# set prefix-list wg-list 2001:db8::10.10
user@host# set prefix-list wg-list 2001:db8::10.22
```

Configuring HTTP Redirect for Local and Remote Redirect Servers

When HTTP requests are made for sites outside the walled garden, CPCD can redirect the traffic to a captive portal for authentication and authorization.

Configure a CPCD service rule that specifies the action to be taken for traffic destined outside the walled garden. This traffic was identified by the walled garden service filter and passed to the service. The action you configure depends on whether you are using a local or a remote HTTP redirect server:

- If you are using a local HTTP redirect server on the router, you specify the redirect action.
- If you are using a remote HTTP redirect server, which resides in a walled garden behind the router, then you cannot simply specify a redirect URL. In this case, the service rule must rewrite the IP destination address for the traffic. The new destination address is the address of the remote HTTP redirect server. The remote server then supplies a redirect URL to send the traffic to a captive portal.

The CPCD service is associated with a service interface by a service set. Both the service set and the walled garden service filter are applied to a statically configured interface.

1. Access the CPCD service configuration level.

```
[edit services]
user@host# edit captive-portal-content-delivery
```

2. Create a rule to apply to traffic destined outside the walled garden.

```
[edit services captive-portal-content-delivery]
edit rule name
```

3. Specify that the rule applies to incoming traffic.

```
[edit services captive-portal-content-delivery rule name]
user@host# set match-direction input
```

4. Define a rule term for CPCD to apply an action to HTTP traffic. Because the walled garden is configured as a service filter, the traffic is already filtered to be HTTP traffic before being sent to the service.

- For a local HTTP redirect server, provide the redirect URL, which is the URL of the captive portal with the original URL (outside the walled garden) appended:

```
[edit services captive-portal-content-delivery rule name]
user@host# set term name then redirect redirect-url/url=%dest-url%
```

- For a remote HTTP redirect server, provide the destination address of the remote server:

```
[edit services captive-portal-content-delivery rule name]
user@host# set term name then rewrite destination-address remote-server-address
```

For example, in the following configuration for a local server, the CPCD service rule redir-svc redirects traffic to a captive portal, `http://www.portal.example.com`. The original URL entered by the subscriber is appended to the redirect URL.

```
user@host# edit services captive-portal-content-delivery
user@host# edit rule redir-svc
user@host# set match-direction input
user@host# set term redir1 then redirect http://www.portal.example.com/url=%dest-url%
```

The following configuration for a remote server creates CPCD service rule rewr-svc that rewrites the original destination address to the address of the remote server, 192.0.2.230.

```
user@host# edit services captive-portal-content-delivery
user@host# edit rule rewr-svc
user@host# set match-direction input
user@host# set term rewr1 then rewrite destination-address 192.0.2.230
```

Configuring the Service Profile and the Service Set to Associate the Service Profile with a Service Interface

Service sets define one or more services to be performed by the MS-MPC/MS-MIC, or the MX-SPC3 services card if you have enabled NEXT Gen Services on the MX Series router. For HTTP redirect services, you define a CPCD service profile that includes CPCD rules. The service set applies the CPCD service profile to a specific service interface.

1. Create the service profile.

```
[edit services captive-portal-content-delivery]
user@host# edit profile name
```

2. Specify one or more CPCD rules for the service profile.

```
[edit services captive-portal-content-delivery profile name]
user@host# set cpcd-rules rule-name
```

3. Create the service set.

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

4. Specify the CPCD service profile.

```
[edit services service-set name]
user@host# set captive-portal-content-delivery-profile name
```

5. Specify the service interface.

```
[edit services service-set name]
user@host# set interface-service service-interface interface-name
```

For example, the following configuration creates CPCD service profile redir-prof, which references the CPCD rule redir-svc. Service set ss2 associates the CPCD service profile redir-prof with the service interface ms-5/0/0.

```
[edit services captive-portal-content-delivery]
user@host# edit profile redir-prof
user@host# set cpcd-rules redir-svc
[edit services]
user@host# edit service-set sset2
user@host# set captive-portal-content-delivery-profile redir-prof
user@host# set interface-service-service-interface ms-5/0/0
```

Attaching a CPCD Service Set and Service Filter to a Logical Interface

To use the HTTP redirect services, you must attach the CPCD service set to a logical interface. If the walled garden is configured as a service filter, then you must attach it to the same interface as the service set. Traffic arriving on and leaving that interface is filtered by the service filter. Traffic identified for servicing is sent to the MS-MPC, or to the MX-SPC3 services card if you have enabled Next Gen Services on the MX Series router, and the CPCD profile is applied at the service interface.

1. Configure the logical interface.

```
[edit interfaces interface-name]
user@host# edit unit logical-unit-number
```

2. Configure the address family.

```
[edit interfaces interface-name unit logical-unit-number]
user@host# edit family family
```

3. Configure the interface address.

```
[edit interfaces interface-name unit logical-unit-number family family]
user@host# set address address
```

4. Attach the service set and service filter to the interface.

```
[edit interfaces interface-name unit logical-unit-number family family]
user@host# set service input service-set set-name service-filter filter-name
user@host# set service output service-set set-name service-filter filter-name
```

For example, the following configuration attaches service set sset2 and service filter walled-v4 to ge-2/0/1.0 for the IPv4 address family. It assigns an address to the logical interface. The service set and filter are both applied to the interface input and output.

```
user@host# edit interfaces ge-2/0/1 unit 0 family inet
user@host# set address 203.0.113.5
user@host# set service input service-set sset2 service-filter walled-v4
user@host# set service output service-set sset2 service-filter walled-v4
```

Installing a Service Package for CPCD Service

To use CPCD services on an MS-MPC/MS-MIC, or on an MX-SPC3 services card if you have enabled Next Gen Services on the MX Series router, you configure a service interface on the MS-MIC or MX-SPC3. You must install the required service package on each MS-MIC that has a service interface or on theMX-SPC3 services card.

1. Configure Junos OS to support a service package on a service interface on an MX Series 5G Universal Routing Platform with MS-MPCs or anMX-SPC3 services card.

```
[edit]
user@host# edit chassis fpc slot-number pic number adaptive-services service-package
```

2. Configure the CPCD service package to run on the PIC. When the `extension-provider` statement is first configured, the PIC reboots.

NOTE: Static MS-MPC-based or MX-SPC3 services card-based CPCD requires the CPCD service package (jservices-cpcd).

```
[edit chassis fpc slot-number pic number adaptive-services service-package]
user@host# set extension-provider package jservices-cpcd
```

3. (Optional) Enable PIC system logging to record or view system log messages on the PIC. You can specify one or more facilities, each at a configurable severity level.

```
[edit chassis fpc 1 pic 0 adaptive-services service-package]
user@host# set extension-provider syslog facility severity
```

For example, the following configuration loads the CPCD services package on the MS-MPC in chassis slot 1 and the MS-MIC in slot 0 of the MPC. System log messages are generated for any daemon and for local external applications at all severity levels.

```
user@host# edit chassis fpc 1 pic 0 adaptive-services service-package
[edit chassis fpc 1 pic 0 adaptive-services service-package]
user@host# set extension-provider package jservices-cpcd
[edit chassis fpc 1 pic 0 adaptive-services service-package]
user@host# set extension-provider syslog daemon any
user@host# set extension-provider syslog external any
```

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
19.3R1	Starting in Junos OS Release 19.3R2, static HTTP redirect service provisioning is also supported for MX-SPC3 services card–based captive portals if you have enabled Next Gen Services on the MX Series router.

RELATED DOCUMENTATION

- HTTP Redirect Service Overview | 511
- Remote HTTP Redirect Server Operation Flow | 518
- Local HTTP Redirect Server Operation Flow (MX Series, ACX7100-48L, ACX7332 and ACX7348) | 520
- Configuring MS-MPC-Based or MX-SPC3-Based Converged HTTP Redirect Services | 532
- Configuring Routing Engine-Based, Static HTTP Redirect Services | 544
- Adding Subscriber Information to HTTP Redirect URLs | 570

Configuring MS-MPC-Based or MX-SPC3-Based Converged HTTP Redirect Services

IN THIS SECTION

- Configuring a Walled Garden as a Firewall Service Filter | 533
- Configuring HTTP Redirect for Local and Remote Redirect Servers | 537
- Configuring Parameterization for the Redirect URL | 538
- Configuring the Service Set to Associate the Service Profile with a Service Interface | 540
- Attaching a CPCD Service Set and Service Filter to a Dynamic Logical Interface | 541
- Installing a Service Package for CPCD Service | 543

You can configure converged HTTP redirect services on MS-MPCs/MS-MICs. Starting in Junos OS Release 19.3R1, you can also configure converged HTTP redirect service provisioning on the MX-SPC3 services card if you have enabled Next Gen Services on the MX Series router.

Converged service provisioning separates service definition from service instantiation. After a service is defined, a service can be dynamically instantiated at subscriber login or by using a change of authorization (CoA) mid-session. Service instantiation uses only the name of the defined service, hiding all service details from system operators. Converged service provisioning supports service parameterization, which corresponds to dynamic variables within dynamic profiles.

For converged HTTP redirect services, this means that you define the service and service rules within a dynamic profile. The CPCD service rules are created dynamically based on the variables configured in the dynamic profile.

Optionally, you can choose to parameterize the redirect URL by including defining a `redirect-url` variable in the dynamic profile. The value of the variable is provided by a RADIUS VSA during subscriber bring-up or with a Change of Authorization (CoA) message. This enables you to customize the redirect URLs for each subscriber. You can define a default value for the URL that is used if no value is provided by RADIUS.

You configure the walled garden as a firewall service filter. It filters traffic so that only HTTP traffic destined outside the walled garden is passed to the dynamic service for processing. Just as for static HTTP redirect services, a service profile contains the service rules. You configure a service set outside the dynamic profile to associate the CPCD service profile with a specific ms service interface on an MS-MPC/MS-MIC or a vsp service interface on an MX-SPC3 services card. Within the dynamic profile, you apply the service set and the walled garden service filter to a dynamic interface.

Configuring a Walled Garden as a Firewall Service Filter

When you configure the walled garden as a firewall service filter, traffic that is destined to servers within the walled garden is identified and skipped. Because this traffic does not flow to the line card, handling requirements are reduced.

All other HTTP traffic is destined for addresses outside the walled garden. Because this traffic does not match the filter conditions, it flows to the line card for handling.

You can configure the service filter so that the walled garden contains a single server as the captive portal or a list of servers.

- Configure the walled garden with a single server as the captive portal:

  1. Create the service filter.

```
[edit]
user@host# edit firewall family address-family service-filter filter-name
```

2. Define a filter term to identify and skip processing for traffic to the captive portal.

   a. Specify filter conditions to match traffic that is destined for the captive portal by specifying the destination address of the captive portal and the destination port.

```
[edit firewall family inet service-filter filter-name]
user@host# set term name from destination-address ip-address
user@host# set term name from destination-port port-number
```

   b. Specify that the matching traffic skips processing on the line card.

```
[edit firewall family inet service-filter filter-name]
user@host# set term name then skip
```

3. Define a filter term to identify HTTP traffic from all the traffic that did not match the previous term and send it for processing by CPCD service rules.

   a. Specify one or more HTTP port numbers to match the skipped HTTP traffic.

```
[edit firewall family inet service-filter filter-name]
user@host# set term name from destination-port http-port-number
```

   b. Specify that the matching traffic is processed by a CPCD service.

```
[edit firewall family inet service-filter filter-name]
user@host# set term name then service
```

4. Define a filter term to skip further action for any remaining, non-HTTP traffic.

```
[edit firewall family inet service-filter filter-name]
user@host# set term name then skip
```

For example, the following configuration creates a filter for IPv4 HTTP traffic, walled-v4, with the captive portal on 192.0.2.0. Traffic matching the address is skipped. Nonmatching traffic goes to term

http, where HTTP traffic is picked out of all skipped traffic and sent to be processed according to a CPCD service. Finally, term skip causes all the remaining, non-HTTP traffic to be skipped.

```
[edit]
user@host# edit firewall family inet service-filter walled-v4
[edit firewall family inet service-filter walled-v4]
user@host# set term portal from destination-address 192.0.2.0
user@host# set term portal from destination-port 80
user@host# set term portal then skip
user@host# set term http from destination-port 80
user@host# set term http then service
user@host# set term skip then skip
```

- Configure the walled garden as a list or subnet of servers.

  1. Create the service filter.

     ```
     [edit]
     user@host# edit firewall family address-family service-filter filter-name
     ```

  2. Define a filter term.

     a. Specify filter conditions to match traffic that is destined for any server in the walled garden by specifying a destination prefix list of servers.

     ```
     [edit firewall family inet service-filter filter-name]
     user@host# set term name from destination-prefix-list list-name
     user@host# set term name from destination-port port-number
     ```

     b. Specify that the matching traffic skips processing on the line card.

     ```
     [edit firewall family inet service-filter filter-name]
     user@host# set term name then skip
     ```

  3. Define a filter term to identify HTTP traffic from all the traffic that did not match the previous term and send it for processing by CPCD service rules.

a. Specify one or more HTTP port numbers to match the skipped HTTP traffic.

```
[edit firewall family inet service-filter filter-name]
user@host# set term name from destination-port http-port-number
```

b. Specify that the matching traffic is processed by a CPCD service.

```
[edit firewall family inet service-filter filter-name]
user@host# set term name then service
```

4. Define a filter term to skip further action for any remaining, non-HTTP traffic.

```
[edit firewall family inet service-filter filter-name]
user@host# set term name then skip
```

5. (Optional) Define a prefix list that specifies servers within the walled garden. You can specify a subnet or multiple individual addresses.

```
[edit policy-options]
user@host# set prefix-list list- name ip-address/mask
user@host# set prefix-list list- name ip-address1
user@host# set prefix-list list- name ip-address2
```

For example, the following configuration creates a service filter for IPv6 HTTP traffic, walled-v6-list, with a prefix list, wg-list, that specifies two servers in the walled garden. Filter term portal6 identifies IPv6 traffic that is destined for the walled garden. Nonmatching traffic goes to term http6, where HTTP traffic is picked out of all skipped traffic and sent to be processed according to a CPCD service. Finally, term skip causes all the remaining, non-HTTP traffic to skipped.

```
[edit]
user@host# edit firewall family inet6 service-filter walled-v6-list
user@host# set term portal6 from destination-prefix-list wg-list
user@host# set term portal6 then skip
user@host# set term http6 from destination-port [80 8080]
user@host# set term http6 then service
user@host# set term skip6 then skip
[edit policy-options]
```

```
user@host# set prefix-list wg-list 2001:db8::10.10
user@host# set prefix-list wg-list 2001:db8::10.22
```

Configuring HTTP Redirect for Local and Remote Redirect Servers

When HTTP requests are made for sites outside the walled garden, CPCD can redirect the traffic to a captive portal for authentication and authorization.

Configure a CPCD service rule that specifies the action to be taken for the HTTP traffic identified by the walled garden service filter and passed to the service. The action you configure depends on whether you are using a local or a remote HTTP redirect server:

- If you are using a local HTTP redirect server on the router, you specify the redirect action.
- If you are using a remote HTTP redirect server, which resides in a walled garden behind the router, then you cannot simply specify a redirect URL. In this case, the service rule must rewrite the IP destination address for the traffic. The new destination address is the address of the remote HTTP redirect server. The remote server then supplies a redirect URL to send the traffic to a captive portal.

1. Configure the dynamic profile.

```
[edit]
user@host# edit dynamic-profiles profile-name
```

2. Access the dynamic CPCD service configuration level.

```
[edit dynamic-profiles profile-name]
user@host# edit services captive-portal-content-delivery
```

3. Create a rule to apply to traffic destined outside the walled garden.

```
[edit dynamic-profiles profile-name services captive-portal-content-delivery]
edit rule name
```

4. Specify that the rule applies to incoming traffic.

```
[edit dynamic-profiles profile-name services captive-portal-content-delivery rule name]
user@host# set match-direction input
```

5. Specify the action to take for the matching traffic .Because the walled garden is a service filter, the traffic is already identified as HTTP traffic before being sent to the service.

- For a local HTTP redirect server, provide the redirect URL, which is the URL of the captive portal with the original URL (outside the walled garden) appended:

```
[edit dynamic-profiles profile-name services captive-portal-content-delivery rule name]
user@host# set term name then redirect redirect-url/url=%dest-url%
```

- For a remote HTTP redirect server, provide the destination address of the remote server:

```
[edit dynamic-profiles profile-name services captive-portal-content-delivery rule name]
user@host# set term name then rewrite destination-address remote-server-address
```

For example, in the following configuration for a local server, the dynamic profile http-redir-converged includes the CPCD service rule redir-svc. The rule redirects traffic to a captive portal, `http://www.portal.example.com`. The original URL entered by the subscriber is appended to the redirect URL. The CPCD service profile redir-prof includes the rule, and will later be applied to a service interface by a service set.

```
user@host# edit dynamic-profiles http-redir-converged
user@host# edit services captive-portal-content-delivery
user@host# edit rule redir-svc
user@host# set match-direction input
user@host# set term redir1 then redirect http://www.portal.example.com/url=%dest-url%
```

The following configuration for a remote server creates CPCD service rule rewr-svc that rewrites the original destination address to the address of the remote server, 192.0.2.230.

```
user@host# edit dynamic-profiles http-redir-converged
user@host# edit services captive-portal-content-delivery
user@host# edit rule rewr-svc
user@host# set match-direction input
user@host# set term rewr1 then rewrite destination-address 192.0.2.230
```

Configuring Parameterization for the Redirect URL

You can optionally choose to parameterize the redirect URL and the rewrite destination address by specifying user-defined variables in the dynamic profile. Parameterizing means that URL or address becomes a dynamic variable. The value is provided by RADIUS when the subscriber is authenticated or when a CoA is received. Consequently, you can use the RADIUS attributes to provide different URLs or destination addresses for different subscribers.

1. Configure the dynamic profile.

```
[edit]
user@host# edit dynamic-profiles profile-name
```

2. Access the custom variable configuration level.

```
[edit dynamic-profiles profile-name]
user@host# edit variables
```

3. Define the variable for the redirect URL, the rewrite destination address, or both. Specify that the value for the dynamic variable is provided by an external server, typically RADIUS.

NOTE: You can name the variables anything you like, but names like redirect-url and rewrite-da make the purpose clear.

```
[edit dynamic-profiles profile-name variables]
set variable-name mandatory
```

4. In the CPCD rule, specify the variable by prepending a dollar sign ($) to the variable name.
   - For a local HTTP redirect server, provide the redirect variable:

```
[edit dynamic-profiles profile-name services captive-portal-content-delivery rule name]
user@host# set term name then redirect $variable-name
```

   - For a remote HTTP redirect server, provide the destination address variable:

```
[edit dynamic-profiles profile-name services captive-portal-content-delivery rule name]
user@host# set term name then rewrite $variable-name
```

For example, the following configuration shows two user-defined variables, redirect-url and rewrite-da that require externally provided values when they are instantiated. CPCD service rule redir1 specifies traffic is redirected to $redirect-url. CPCD service rule rewr1 specifies that the destination address for the traffic is rewritten to $rewrite-da.

```
user@host# edit dynamic-profiles http-redir-converged
user@host# edit variables
```

```
user@host# set redirect-url mandatory
user@host# set rewrite-da mandatory
user@host# edit services captive-portal-content-delivery
user@host# edit rule redir-svc
user@host# set match-direction input
user@host# set term redir1 then redirect $redirect-url
user@host# edit rule rewr-svc
user@host# set match-direction input
user@host# set term rewr1 then rewrite $rewrite-da
```

Configuring the Service Set to Associate the Service Profile with a Service Interface

Service sets define one or more services to be performed by the MS-MPC/MS-MIC, or by the MX-SPC3 services card if you have enabled Next Gen Services on the MX Series router. For HTTP redirect services, you define a CPCD service profile that includes CPCD rules. The service set applies the CPCD service profile to a specific service interface.

1. Create the service profile.

```
[edit services captive-portal-content-delivery]
user@host# edit profile name
```

2. Specify one or more CPCD rules configured in the CPCD dynamic profile for the service profile.

```
[edit services captive-portal-content-delivery profile name]
user@host# set cpcd-rules rule-name
```

3. Specify that this is a converged CPCD service.

```
[edit services captive-portal-content-delivery profile name]
user@host# set dynamic
```

4. Create the service set.

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

5. Specify the CPCD service profile.

```
[edit services service-set name]
user@host# set captive-portal-content-delivery-profile name
```

6. Specify the service interface.

```
[edit services service-set name]
user@host# set interface-service service-interface interface-name
```

For example, the following configuration creates CPCD service profile redir-prof, which references the CPCD rule redir-svc. Service set cvgd associates the CPCD service profile rewr-prof with the service interface ms-2/0/0.

```
[edit services captive-portal-content-delivery]
user@host# edit profile redir-prof
user@host# set cpcd-rules redir-svc
user@host# set dynamic
[edit services]
user@host# edit service-set cvgd
user@host# set captive-portal-content-delivery-profile redir-prof
user@host# set interface-service service-interface ms-2/0/0
```

Attaching a CPCD Service Set and Service Filter to a Dynamic Logical Interface

To use the HTTP redirect services, you must attach the CPCD service set to a logical interface. Because the walled garden is configured as a service filter, you must attach it to the same interface as the service set. Traffic arriving on and leaving that interface is filtered by the service filter. Traffic identified for servicing is sent to the MS-MPC or MX-SPC3 service interface where the CPCD profile is applied.

NOTE: This procedure shows only elements of the dynamic profile configuration that are specific to the converged services configuration. The complete dynamic profile depends on your use case.

1. Configure the dynamic profile.

```
[edit]
user@host# edit dynamic-profiles profile-name
```

2. Configure the dynamic physical interface.

```
[edit dynamic-profiles profile-name]
user@host# edit interfaces $junos-interface-ifd-name
```

3. Configure the dynamic logical interface.

```
[edit dynamic-profiles profile-name interfaces $junos-interface-ifd-name]
user@host# edit unit $junos-underlying-interface-unit
```

4. Configure the address family.

```
[edit dynamic-profiles profile-name interfaces $junos-interface-ifd-name unit $junos-
underlying-interface-unit]
user@host# edit family family
```

5. Attach the service set and service filter to the interface.

```
[edit dynamic-profiles profile-name interfaces $junos-interface-ifd-name unit $junos-
underlying-interface-unit family family]
user@host# set service input service-set set-name service-filterfilter-name
user@host# set service output service-set set-name service-filter filter-name
```

For example, the following configuration creates the dynamic profile http-redir-converged. It specifies predefined variables to create the dynamic physical and logical interfaces in the IPv4 address family. The profile attaches service set cvgd and service filter walled-v4 to the dynamic logical interface when it is created at subscriber login. The service set and filter are both applied to the interface input and output.

```
user@host# edit dynamic-profiles http-redir-converged
user@host# edit interfaces $junos-interface-ifd-name
user@host# edit unit $junos-underlying-interface-unit
user@host# edit family inet
user@host# set service input service-set cvgd service-filter walled-v4
user@host# set service output service-set cvgd service-filter walled-v4
```

Installing a Service Package for CPCD Service

To use CPCD services on an MS-MPC/MS-MIC, or on an MX-SPC3 services card if you have enabled USF on the MX Series router, you configure a service interface on the MS-MIC or MX-SPC3. You must install the required service packages on each MS-MIC that has a service interface or on an MX-SPC3.

1. Configure Junos OS to support a service package on a service interface on an MX Series 5G Universal Routing Platform with MS-MPCs or on an MX-SPC3 services card for Next Gen Services.

```
[edit]
user@host# edit chassis fpc slot-number pic number adaptive-services service-package
```

2. Configure the required service packages to run on the PIC. When the `extension-provider`

> **NOTE**: Converged services MS-MPC-based or MX-SPC3-based CPCD requires both the CPCD service package (jservices-cpcd) and the mobile subscriber service package (jservices-mss).

```
[edit chassis fpc slot-number pic number adaptive-services service-package]
user@host# set extension-provider package jservices-cpcd,jservices-mss
```

3. (Optional) Enable PIC system logging to record or view system log messages on the PIC. You can specify one or more facilities, each at a configurable severity level.

```
[edit chassis fpc 1 pic 0 adaptive-services service-package]
user@host# set extension-provider syslog facility severity
```

For example, the following configuration loads the CPCD services package and the mobile subscriber services package on the MS-MPC in chassis slot 1 and the MS-MIC in slot 0 of the MPC. System log messages are generated for any daemon and for local external applications at all severity levels.

```
user@host# edit chassis fpc 1 pic 0 adaptive-services service-package
[edit chassis fpc 1 pic 0 adaptive-services service-package]
user@host# set extension-provider package jservices-cpcd,jservices-mss
[edit chassis fpc 1 pic 0 adaptive-services service-package]
user@host# set extension-provider syslog daemon any
user@host# set extension-provider syslog external any
```

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
19.3R1	Starting in Junos OS Release 19.3R1, you can also configure converged HTTP redirect service provisioning on the MX-SPC3 services card if you have enabled Next Gen Services on the MX Series router.

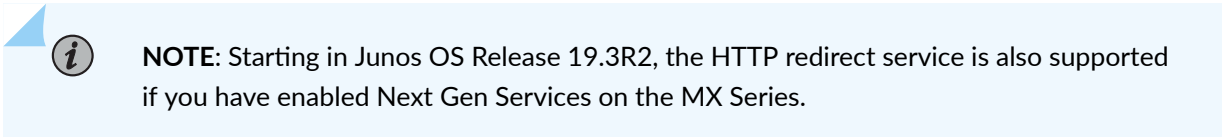
RELATED DOCUMENTATION

- *Dynamic Profiles Overview*
- *Dynamic Variables Overview*
- *Junos OS Predefined Variables*
- *User-Defined Variables*
- HTTP Redirect Service Overview | **511**
- Remote HTTP Redirect Server Operation Flow | **518**
- Local HTTP Redirect Server Operation Flow (MX Series, ACX7100-48L, ACX7332 and ACX7348) | **520**
- Configuring MS-MPC-Based or MX-SPC3-Based Static HTTP Redirect Services | **522**
- Configuring Routing Engine-Based, Converged HTTP Redirect Services | **559**
- Adding Subscriber Information to HTTP Redirect URLs | **570**

Configuring Routing Engine-Based, Static HTTP Redirect Services

IN THIS SECTION

- Configuring a Walled Garden as a Firewall Service Filter | **545**
- Configuring HTTP Redirect for Local and Remote Redirect Servers | **549**
- Configuring the Service Profile and the Service Set to Associate the Service Profile with a Service Interface | **551**
- Attaching a CPCD Service Set and Service Filter to a Logical Interface | **552**
- Inserting GET Header Tags That the HTTP Server Can Use to Control Content Access | **553**



NOTE: Starting in Junos OS Release 19.3R2, the HTTP redirect service is also supported if you have enabled Next Gen Services on the MX Series.

You can configure HTTP redirect services on the Routing Engine as an alternative to using an MS-MPC/MS-MIC or MX-SPC3 services card. You configure the walled garden as a firewall service filter. A walled garden is a group of servers that provide subscriber access to sites within the walled garden without requiring reauthorization through a captive portal. The walled garden service filter identifies traffic destined for the walled garden and traffic destined outside the walled garden. Only HTTP traffic destined outside the walled garden is sent to the Routing Engine for processing by the HTTP redirect service. The CPCD service is associated with a service interface on the Routing Engine by means of a service set.

The service interfaces on the Routing Engine are identified with an si- prefix (for example, si-1/1/0). The si- interface processes all redirect and rewrite traffic and services for the Routing Engine. The si- interface must be operational with a status of up to enable and activate the captive portal content delivery (CPCD) service. After the CPCD service is enabled, any change in the operational state of the si- interface does not affect existing CPCD services.

The CPCD service sends the subscriber HTTP request traffic that is not destined for the walled garden to a redirect server, which responds with a redirect URL. The redirect URL sends traffic to a captive portal instead of to the unauthorized external site. The captive portal provides authentication and authorization services for the redirected subscribers before granting them access to protected servers outside of the walled garden.

The redirect server can be local or remote:

- Local redirect server—Resides on the router and redirects subscriber traffic to a captive portal inside a walled garden.
- Remote redirect server—Resides on a device such as a policy server inside a walled garden behind the router. The destination address for the subscriber's HTTP traffic is rewritten to the address of the remote redirect server. The remote server redirects subscriber traffic to a captive portal inside that walled garden.

Configuring a Walled Garden as a Firewall Service Filter

When you configure the walled garden as a firewall service filter, traffic that is destined to servers within the walled garden is identified and skipped. All other HTTP traffic is destined for addresses outside the walled garden. Because this traffic does not match the filter conditions, it flows to the Routing Engine for handling.

NOTE: On Junos OS Evolved, when you configure skip action within a service filter, it results in dropping the skipped packets.

You can configure the service filter so that the walled garden contains a single server as the captive portal or a list of servers.

- Configure the walled garden with a single server as the captive portal:

  1. Create the service filter.

     ```
     [edit]
     user@host# edit firewall family address-family service-filter filter-name
     ```

  2. Define a filter term to identify and skip processing for traffic to the captive portal.

     a. Specify filter conditions to match traffic that is destined for the captive portal by specifying the destination address of the captive portal and the destination port.

     ```
     [edit firewall family inet service-filter filter-name]
     user@host# set term name from destination-address ip-address
     user@host# set term name from destination-port port-number
     ```

     b. Specify that the matching traffic skips processing on the line card.

     ```
     [edit firewall family inet service-filter filter-name]
     user@host# set term name then skip
     ```

  3. Define a filter term to identify HTTP traffic from all the traffic that did not match the previous term and send it for processing by CPCD service rules.

     a. Specify one or more HTTP port numbers to match the skipped HTTP traffic.

     ```
     [edit firewall family inet service-filter filter-name]
     user@host# set term name from destination-port http-port-number
     ```

b. Specify that the matching traffic is processed by a CPCD service.

```
[edit firewall family inet service-filter filter-name]
user@host# set term name then service
```

4. Define a filter term to skip further action for any remaining, non-HTTP traffic.

```
[edit firewall family inet service-filter filter-name]
user@host# set term name then skip
```

For example, the following configuration creates a filter for IPv4 HTTP traffic, walled-v4, with the captive portal on 192.0.2.0. Processing is skipped for traffic matching the address; the traffic is sent to the captive portal. Nonmatching traffic goes to term http, where HTTP traffic is picked out of all skipped traffic and sent to be processed according to a CPCD service. Finally, term skip causes all the remaining, non-HTTP traffic to be skipped.

```
[edit]
user@host# edit firewall family inet service-filter walled-v4
[edit firewall family inet service-filter walled-v4]
user@host# set term portal from destination-address 192.0.2.0
user@host# set term portal from destination-port 80
user@host# set term portal then skip
user@host# set term http from destination-port 80
user@host# set term http then service
user@host# set term skip then skip
```

- Configure the walled garden as a list or subnet of servers.

1. Create the service filter.

```
[edit]
user@host# edit firewall family address-family service-filter filter-name
```

2. Define a filter term.

a. Specify filter conditions to match traffic that is destined for any server in the walled garden by specifying a destination prefix list of servers.

```
[edit firewall family inet service-filter filter-name]
user@host# set term name from destination-prefix-list list-name
user@host# set term name from destination-port port-number
```

NOTE: On Junos OS Evolved, you can create more than one prefix-list, each can have a maximum of three or four prefixes.

b. Specify that the matching traffic skips processing on the line card.

```
[edit firewall family inet service-filter filter-name]
user@host# set term name then skip
```

3. Define a filter term to identify HTTP traffic from all the traffic that did not match the previous term and send it for processing by CPCD service rules.

a. Specify one or more HTTP port numbers to match the skipped HTTP traffic.

```
[edit firewall family inet service-filter filter-name]
user@host# set term name from destination-port http-port-number
```

b. Specify that the matching traffic is processed by a CPCD service.

```
[edit firewall family inet service-filter filter-name]
user@host# set term name then service
```

4. Define a filter term to skip further action for any remaining, non-HTTP traffic.

```
[edit firewall family inet service-filter filter-name]
user@host# set term name then skip
```

5. (Optional) Define a prefix list that specifies servers within the walled garden. You can specify a subnet or multiple individual addresses.

```
[edit policy-options]
user@host# set prefix-list list- name ip-address/mask
user@host# set prefix-list list- name ip-address1
user@host# set prefix-list list- name ip-address2
```

For example, the following configuration creates a filter for IPv6 HTTP traffic, walled-v6-list, with a prefix list, wg-list, that specifies two servers in the walled garden. Filter term portal6 identifies IPv6 traffic that is destined for the walled garden. Nonmatching traffic goes to term http6, where HTTP traffic is picked out of all skipped traffic and sent to be processed according to a CPCD service. Finally, term skip6 causes all the remaining, non-HTTP traffic to skipped.

```
[edit]
user@host# edit firewall family inet6 service-filter walled-v6-list
user@host# set term portal6 from destination-prefix-list wg-list
user@host# set term portal6 then skip
user@host# set term http6 from destination-port [80 8080]
user@host# set term http6 then service
user@host# set term skip6 then skip
[edit policy-options]
user@host# set prefix-list wg-list 2001:db8::10.10
user@host# set prefix-list wg-list 2001:db8::10.22
```

Configuring HTTP Redirect for Local and Remote Redirect Servers

When HTTP requests are made for sites outside the walled garden, CPCD can redirect the traffic to a captive portal for authentication and authorization.

Configure a CPCD service rule that specifies the action to be taken for traffic destined outside the walled garden. This traffic was identified by the walled garden service filter and passed to the service, or identified and accepted by the walled garden service rule. The action you configure depends on whether you are using a local or a remote HTTP redirect server:

- If you are using a local HTTP redirect server on the router, you specify the redirect action.
- If you are using a remote HTTP redirect server, which resides in a walled garden behind the router, then you cannot simply specify a redirect URL. In this case, the service rule must rewrite the IP destination address for the traffic. The new destination address is the address of the remote HTTP redirect server. The remote server then supplies a redirect URL to send the traffic to a captive portal.

The CPCD service is associated with a service interface by a service set. Both the service set and the walled garden service filter are applied to a statically configured interface.

1. Access the CPCD service configuration level.

```
[edit services]
user@host# edit captive-portal-content-delivery
```

2. Create a rule to apply to traffic destined outside the walled garden.

```
[edit services captive-portal-content-delivery]
edit rule name
```

3. Specify that the rule applies to incoming traffic.

```
[edit services captive-portal-content-delivery rule name]
user@host# set match-direction input
```

4. Define a rule term for CPCD to apply an action to HTTP traffic. Because the walled garden is configured as a service filter, the traffic is already filtered to be HTTP traffic before being sent to the service.
   - For a local HTTP redirect server, provide the redirect URL, which is the URL of the captive portal with the original URL (outside the walled garden) appended:

     ```
     [edit services captive-portal-content-delivery rule name]
     user@host# set term name then redirect redirect-url/url=%dest-url%
     ```

   - For a remote HTTP redirect server, provide the destination address of the remote server:

     ```
     [edit services captive-portal-content-delivery rule name]
     user@host# set term name then rewrite destination-address remote-server-address
     ```

For example, in the following configuration for a local server, the CPCD service rule redir-svc redirects traffic to a captive portal, `http://www.portal.example.com`. The original URL entered by the subscriber is appended to the redirect URL.

```
user@host# edit services captive-portal-content-delivery
user@host# edit rule redir-svc
```

```
user@host# set match-direction input
user@host# set term redir1 then redirect http://www.portal.example.com/url=%dest-url%
```

The following configuration for a remote server creates CPCD service rule rewr-svc that rewrites the original destination address to the address of the remote server, 192.0.2.230.

```
user@host# edit services captive-portal-content-delivery
user@host# edit rule rewr-svc
user@host# set match-direction input
user@host# set term rewr1 then rewrite destination-address 192.0.2.230
```

Configuring the Service Profile and the Service Set to Associate the Service Profile with a Service Interface

Service sets define one or more services to be performed by the Routing Engine. For HTTP redirect services, you define a CPCD service profile that includes CPCD rules. The service set applies the CPCD service profile to a specific service interface.

1. Create the service profile.

```
[edit services captive-portal-content-delivery]
user@host# edit profile name
```

2. Specify one or more CPCD rules for the service profile.

```
[edit services captive-portal-content-delivery profile name]
user@host# set cpcd-rules rule-name
```

3. Create the service set.

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

4. Specify that the service set is for Routing Engine–Based CPCD.

```
[edit services service-set name]
user@host# set service-set-options routing-engine-services
```

5. Specify the CPCD service profile.

```
[edit services service-set name]
user@host# set captive-portal-content-delivery-profile name
```

6. Specify the service interface.

```
[edit services service-set name]
user@host# set interface-service service-interface interface-name
```

For example, the following configuration creates CPCD service profile redir-prof, which references the CPCD rule redir-svc. Service set ss2 is specified as being for Routing-Engine-based CPCD. The set associates the CPCD service profile redir-prof with the service interface si-4/0/0.

```
[edit services captive-portal-content-delivery]
user@host# edit profile redir-prof
user@host# set cpcd-rules redir-svc
[edit services]
user@host# edit service-set ss2
user@host# set service-set-options routing-engine-service
user@host# set captive-portal-content-delivery-profile redir-prof
user@host# set interface-service-service-interface si-4/0/0
```

Attaching a CPCD Service Set and Service Filter to a Logical Interface

To use the HTTP redirect services, you must attach the CPCD service set to a logical interface. If the walled garden is configured as a service filter, then you must attach it to the same interface as the service set. Traffic arriving on and leaving that interface is filtered by the service filter. Traffic identified for servicing is sent to the Routing Engine service interface where the CPCD profile is applied.

1. Enable inline services and specify a bandwidth.

```
[edit chassis fpc slot-number pic number]
user@host# set inline-services bandwidth bandwidth
```

2. Configure the logical inline services interface.

```
[edit interfaces interface-name]
user@host# edit unit logical-unit-number
```

3. Configure the address family.

```
[edit interfaces interface-name unit logical-unit-number]
user@host# edit family family
```

4. Attach the service set and service filter to the interface.

   - Static interface:

```
[edit interfaces interface-name unit logical-unit-number family family]
user@host# set service input service-set set-name service-filter filter-name
user@host# set service output service-set set-name service-filter filter-name
```

   - Dynamic interface

```
[edit dynamic-profiles profile-name interfaces interface-name unit logical-unit-number
family family]
user@host# set service input service-set set-name service-filter filter-name
user@host# set service output service-set set-name service-filter filter-name
```

For example, the following configuration enables inline services on the line card in chassis slot 4 and on the MIC in slot 0 of the line card. It assigns an address to the logical interface. Then it attaches service set sset2 and service filter walled-v4 to ge-2/0/1.0 for the IPv4 address family. The service set and filter are both applied to the interface input and output.

```
user@host# edit chassis fpc 4 pic 0 inline-services bandwidth 1g
user@host# edit interfaces ge-2/0/1 unit 0 family inet
user@host# set address 203.0.113.5
user@host# set service input service-set sset2 service-filter walled-v4
user@host# set service output service-set sset2 service-filter walled-v4
```

Inserting GET Header Tags That the HTTP Server Can Use to Control Content Access

In some cases you might want your HTTP server to determine whether to allow users to access content. Starting in Junos OS Release 19.1, you can configure Routing Engine-based, static HTTP redirect service filters to specify tags that the Routing Engine inserts in to the packet header of HTTP GET messages for this purpose. You can insert tags for the router hostname or the subscriber's MAC address, IPv4 address, or IPv6 address.

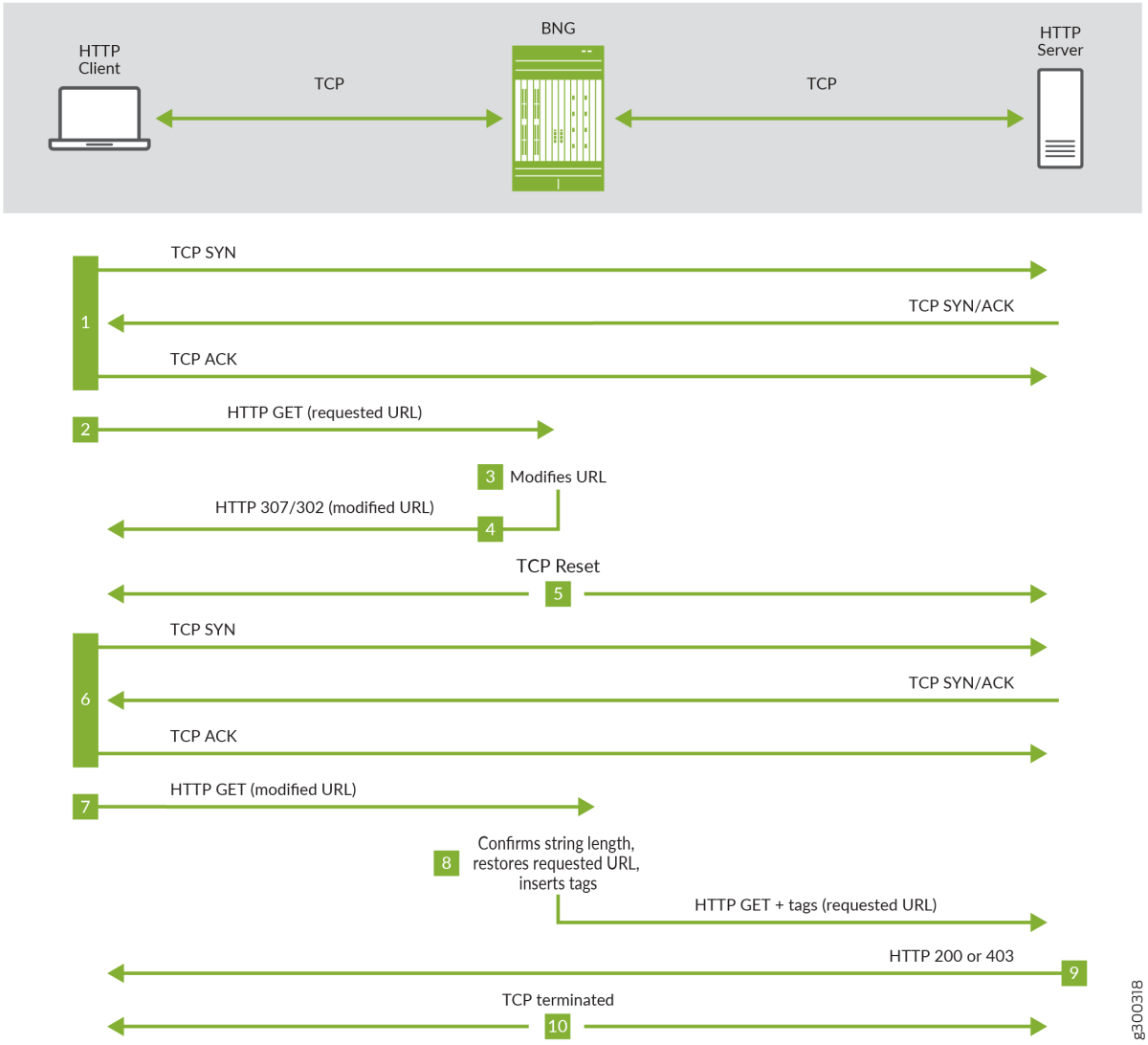
The following steps correspond to Figure 10 on page 555.

1. The user's device, the HTTP client, performs a TCP handshake sequence with the HTTP server.
2. When the handshake is successful, the client sends an HTTP GET with the URL requested by the user.
3. The Routing Engine modifies that URL by concatenating a string of random characters enclosed by /$ and $/. The string length matches the combined length of the tags that will be inserted later. The string serves as an identifier when returned by the client.

   Suppose the length of the tags to be inserted is 30 characters, and the requested URL is http://192.51.100.20/test.html. The Routing Engine returns the URL modified with a string of 30 random characters as in the following example:

   http://192.51.100.20/test.html/$IIGSbVdNDTDvnJFIAyoysXwVJawoYj$/
4. The Routing Engine sends the modified URL with a status code of 302 (Found) or 307 (Temporary Redirect). The code sent depends on the version of HTTP being used and the version of Junos OS on the BNG. Both codes indicate to the client that the access request needs to be resent with the modified URL.
5. The Routing Engine resets the TCP connection with the client and the server.
6. The client performs a TCP handshake with the HTTP server for a modified URL.
7. The client sends an HTTP GET with the modified URL.
8. The Routing Engine checks whether the length of the concatenated string is the same as it sent to the client.
   - If the length is correct, it strips the URL back to the original requested URL, inserts the tags into the GET header, and forwards the GET to the HTTP server. If configured, the GET can be optionally forwarded to a redirect URL instead of the original requested server.
   - If the length is not correct, the Routing Engine drops the packet and increments the drop counter.
9. The HTTP server evaluates the GET message and sends a response to the client with a status code of 200 (OK) if it grants access or 403 (Forbidden) if the request is rejected.
10. The Routing Engine terminates the TCP connection with the client and the server.

Figure 10: Tag Insertion for HTTP Redirect Message Flow.



NOTE: Tags are inserted into the header in the same order as they are configured. The tag name is case-sensitive so that `tag ABCD` and `tag abcd` are processed as different names.

To configure tags to be inserted in GET headers:

1. Access the CPCD service configuration level.

```
[edit services]
user@host# edit captive-portal-content-delivery
```

2. Create a rule to apply to traffic destined outside the walled garden.

```
[edit services captive-portal-content-delivery]
edit rule name
```

3. Specify that the rule applies to incoming traffic.

```
[edit services captive-portal-content-delivery rule name]
user@host# set match-direction input
```

4. (Optional) Specify one or more destination addresses to filter traffic for tagging.

```
[edit services captive-portal-content-delivery rule name]
user@host# set term name from destination-address address
```

NOTE: Alternatively, you can specify destination addresses for identifying traffic in the firewall service filter.

5. Define a rule term for CPCD to apply an action to HTTP traffic. Because the walled garden is configured as a service filter, the traffic is already filtered to be HTTP traffic before being sent to the service.

```
[edit services captive-portal-content-delivery rule name]
user@host# set term name then insert tag tag-name tag-value tag-value
user@host# set term name then insert tag tag-name tag-value tag-value
```

For example, the following configuration creates a service rule, insert-rule, that matches traffic on the input interface. Term t1 inserts two tags, x-mac-addr with the subscriber's MAC address and x-sub-ip with the value of the subscriber's IPv4 address.

```
[edit]
user@host# edit services captive-portal-content-delivery rule insert-rule
user@host# set match-direction input
```

```
user@host# set term t1 then insert tag x-mac-addr tag-value subscriber-mac-addr
user@host# set term t1 then insert tag x-sub-ip tag-value subscriber-ip
```

In the following sample rule, only traffic with a destination address that matches 198.51.100.50 or 198.51.100.75 is tagged. Tags are inserted for the subscriber's IP address and the hostname of the router. A second term in the rule provides a redirect URL where the traffic is forwarded instead of being sent to the original requested URL.

```
user@host# edit services captive-portal-content-delivery
user@host# set match-direction input
user@host# set rule tag-redirect term t1 from destination-address 198.51.100.50
user@host# set rule tag-redirect term t1 from destination-address 198.51.100.75
user@host# set rule tag-redirect term t1 then insert tag x-sub-ip tag-value subscriber-ip
user@host# set rule tag-redirect term t1 then insert tag x-hostname tag-value hostname
user@host# set rule tag-redirect term t2 then redirect http://www.portal.example.com
user@host# set profile http-insert-redirect cpcd-rules tag-redirect
```

As with any CPCD service rules for Routing-Engine-Based HTTP redirect, you must include the rules in a CPCD service profile, then use a CPCD service set to associate the profile with an inline service interface. The Routing Engine uses the rules to process HTTP traffic passed by a service filter on the same logical interface as the service set.

Consider the following sample configuration. The tag-redirect rule is defined to match traffic on the input interface and then insert two tags in the GET header, the value of the subscriber's IP address and the hostname of the router. The rule then provides a redirect URL for the tagged traffic. The CPCD service profile http-insert-redirect is defined to include this rule.

```
user@host# edit services captive-portal-content-delivery
user@host# set match-direction input
user@host# set rule tag-redirect term t1 then insert tag x-sub-ip tag-value subscriber-ip
user@host# set rule tag-redirect term t1 then insert tag x-hostname tag-value hostname
user@host# set rule tag-redirect term t2 then redirect http://www.portal.example.com
user@host# set profile http-insert-redirect cpcd-rules tag-redirect
```

The service set sset1 is defined as being for Routing Engine-based CPCD. It applies the CPCD service profile to an inline service interface.

```
user@host# edit services service-set sset1
user@host# set service-set-options routing-engine-services
```

```
user@host# set captive-portal-content-delivery-profile http-insert-redirect
user@host# set interface-service service-interface si-1/1/0
```

The service filter walled-tag identifies and acts on three kinds of traffic: HTTP traffic to send to the walled garden at 192.0.2.100, HTTP traffic destined for 198.51.100.50 to go to service processing, and all other traffic to be skipped. This is an example of matching a destination address in the service filter instead of in the service rule.

```
user@host# edit firewall family inet service-filter walled-tag
user@host# set term portal from destination-address 192.0.2.100
user@host# set term portal from destination-port 80
user@host# set term portal then skip
user@host# set term http-tag from destination-address 198.51.100.50
user@host# set term http-tag from destination-port 80
user@host# set term http-tag then service
user@host# set term skip then skip
```

The service-set sset1 and service filter walled-tag are applied to a logical interface.

```
user@host# edit chassis fpc 4 pic 0 inline-services bandwidth 1g
user@host# edit interfaces ge-2/0/1 unit 0 family inet
user@host# set address 203.0.113.5
user@host# set service input service-set sset1 service-filter walled-tag
user@host# set service output service-set sset1 service-filter walled-tag
```

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
19.1	Starting in Junos OS Release 19.1, you can configure Routing Engine-based, static HTTP redirect service filters to specify tags that the Routing Engine inserts in to the packet header of HTTP GET messages for this purpose.

RELATED DOCUMENTATION

- HTTP Redirect Service Overview | 511
- Remote HTTP Redirect Server Operation Flow | 518

Local HTTP Redirect Server Operation Flow (MX Series, ACX7100-48L, ACX7332 and ACX7348) | 520

Configuring Routing Engine-Based, Converged HTTP Redirect Services | 559

Configuring MS-MPC-Based or MX-SPC3-Based Static HTTP Redirect Services | 522

Adding Subscriber Information to HTTP Redirect URLs | 570

Configuring Routing Engine-Based, Converged HTTP Redirect Services

IN THIS SECTION

- Configuring a Walled Garden as a Firewall Service Filter | 560
- Configuring HTTP Redirect for Local and Remote Redirect Servers | 563
- Configuring Parameterization for the Redirect URL | 565
- Configuring the Service Set to Associate the Service Profile with a Service Interface | 567
- Attaching a CPCD Service Set and Service Filter to a Dynamic Logical Interface | 568

NOTE: Starting in Junos OS Release 19.3R1, the HTTP redirect service is also supported if you have enabled Next Gen Services on the MX Series.

You can configure converged HTTP redirect services on the Routing Engine as an alternative to using an MS-MPC/MS-MIC or MX-SPC3 services card. Converged service provisioning separates service definition from service instantiation. After a service is defined, a service can be dynamically instantiated at subscriber login or by using a change of authorization (CoA) mid-session. Service instantiation uses only the name of the defined service, hiding all service details from system operators. Converged service provisioning supports service parameterization, which corresponds to dynamic variables within dynamic profiles.

For converged HTTP redirect services, this means that you define the service and service rules within a dynamic profile. The CPCD service rules are created dynamically based on the variables configured in the dynamic profile.

Optionally, you can choose to parameterize the redirect URL by including defining a `redirect-url` variable in the dynamic profile. The value of the variable is provided by a RADIUS VSA during subscriber bring-up or with a Change of Authorization (CoA) message. This enables you to customize the redirect URLs for

each subscriber. You can define a default value for the URL that is used if no value is provided by RADIUS.

You configure the walled garden as a firewall service filter. A walled garden is a group of servers that provide subscriber access to sites within the walled garden without requiring reauthorization through a captive portal. The walled garden service filter identifies traffic destined for the walled garden and traffic destined outside the walled garden. Only HTTP traffic destined outside the walled garden is passed to the dynamic service for processing.

The service interfaces on the Routing Engine are identified with an si- prefix (for example, si-1/1/0). The si- interface processes all redirect and rewrite traffic and services for the Routing Engine. The si- interface must be operational with a status of up to enable and activate the captive portal content delivery (CPCD) service. After the CPCD service is enabled, any change in the operational state of the si- interface does not affect existing CPCD services.

Just as for static HTTP redirect services, a service profile contains the service rules. You configure a service set outside the dynamic profile to associate the CPCD service profile with a specific si service interface on the Routing Engine. Within the dynamic profile, you apply the service set and the walled garden service filter to a dynamic interface.

Configuring a Walled Garden as a Firewall Service Filter

When you configure the walled garden as a firewall service filter, traffic that is destined to servers within the walled garden is identified and skipped. Because this traffic does not flow to the line card, handling requirements are reduced.

All other HTTP traffic is destined for addresses outside the walled garden. Because this traffic does not match the filter conditions, it flows to the line card for handling.

You can configure the service filter so that the walled garden contains a single server as the captive portal or a list of servers.

- Configure the walled garden with a single server as the captive portal:

  1. Create the service filter.

     ```
     [edit]
     user@host# edit firewall family address-family service-filter filter-name
     ```

  2. Define a filter term to identify and skip processing for traffic to the captive portal.

a. Specify filter conditions to match traffic that is destined for the captive portal by specifying the destination address of the captive portal and the destination port.

```
[edit firewall family inet service-filter filter-name]
user@host# set term name from destination-address ip-address
user@host# set term name from destination-port port-number
```

b. Specify that the matching traffic skips processing on the line card.

```
[edit firewall family inet service-filter filter-name]
user@host# set term name then skip
```

3. Define a filter term to identify HTTP traffic from all the traffic that did not match the previous term and send it for processing by CPCD service rules.

a. Specify one or more HTTP port numbers to match the skipped HTTP traffic.

```
[edit firewall family inet service-filter filter-name]
user@host# set term name from destination-port http-port-number
```

b. Specify that the matching traffic is processed by a CPCD service.

```
[edit firewall family inet service-filter filter-name]
user@host# set term name then service
```

4. Define a filter term to skip further action for any remaining, non-HTTP traffic.

```
[edit firewall family inet service-filter filter-name]
user@host# set term name then skip
```

For example, the following configuration creates a filter for IPv4 HTTP traffic, walled-v4, with the captive portal on 192.0.2.0. Traffic matching the address is skipped. Nonmatching traffic goes to term http, where HTTP traffic is picked out of all skipped traffic and sent to be processed according to a CPCD service. Finally, term skip causes all the remaining, non-HTTP traffic to be skipped.

```
[edit]
user@host# edit firewall family inet service-filter walled-v4
```

```
[edit firewall family inet service-filter walled-v4]
user@host# set term portal from destination-address 192.0.2.0
user@host# set term portal from destination-port 80
user@host# set term portal then skip
user@host# set term http from destination-port 80
user@host# set term http then service
user@host# set term skip then skip
```

- Configure the walled garden as a list or subnet of servers.

  1. Create the service filter.

     ```
     [edit]
     user@host# edit firewall family address-family service-filter filter-name
     ```

  2. Define a filter term.

     a. Specify filter conditions to match traffic that is destined for any server in the walled garden by specifying a destination prefix list of servers.

     ```
     [edit firewall family inet service-filter filter-name]
     user@host# set term name from destination-prefix-list list-name
     user@host# set term name from destination-port port-number
     ```

     b. Specify that the matching traffic skips processing on the line card.

     ```
     [edit firewall family inet service-filter filter-name]
     user@host# set term name then skip
     ```

  3. Define a filter term to identify HTTP traffic from all the traffic that did not match the previous term and send it for processing by CPCD service rules.

     a. Specify one or more HTTP port numbers to match the skipped HTTP traffic.

     ```
     [edit firewall family inet service-filter filter-name]
     user@host# set term name from destination-port http-port-number
     ```

b. Specify that the matching traffic is processed by a CPCD service.

```
[edit firewall family inet service-filter filter-name]
user@host# set term name then service
```

4. Define a filter term to skip further action for any remaining, non-HTTP traffic.

```
[edit firewall family inet service-filter filter-name]
user@host# set term name then skip
```

5. (Optional) Define a prefix list that specifies servers within the walled garden. You can specify a subnet or multiple individual addresses.

```
[edit policy-options]
user@host# set prefix-list list- name ip-address/mask
user@host# set prefix-list list- name ip-address1
user@host# set prefix-list list- name ip-address2
```

For example, the following configuration creates a service filter for IPv6 HTTP traffic, walled-v6-list, with a prefix list, wg-list, that specifies two servers in the walled garden. Filter term portal6 identifies IPv6 traffic that is destined for the walled garden. Nonmatching traffic goes to term http6, where HTTP traffic is picked out of all skipped traffic and sent to be processed according to a CPCD service. Finally, term skip causes all the remaining, non-HTTP traffic to skipped.

```
[edit]
user@host# edit firewall family inet6 service-filter walled-v6-list
user@host# set term portal6 from destination-prefix-list wg-list
user@host# set term portal6 then skip
user@host# set term http6 from destination-port [80 8080]
user@host# set term http6 then service
user@host# set term skip6 then skip
[edit policy-options]
user@host# set prefix-list wg-list 2001:db8::10.10
user@host# set prefix-list wg-list 2001:db8::10.22
```

Configuring HTTP Redirect for Local and Remote Redirect Servers

When HTTP requests are made for sites outside the walled garden, CPCD can redirect the traffic to a captive portal for authentication and authorization.

Configure a CPCD service rule that specifies the action to be taken for the HTTP traffic identified by the walled garden service filter and passed to the service. The action you configure depends on whether you are using a local or a remote HTTP redirect server:

- If you are using a local HTTP redirect server on the router, you specify the redirect action.
- If you are using a remote HTTP redirect server, which resides in a walled garden behind the router, then you cannot simply specify a redirect URL. In this case, the service rule must rewrite the IP destination address for the traffic. The new destination address is the address of the remote HTTP redirect server. The remote server then supplies a redirect URL to send the traffic to a captive portal.

1. Configure the dynamic profile.

```
[edit]
user@host# edit dynamic-profiles profile-name
```

2. Access the dynamic CPCD service configuration level.

```
[edit dynamic-profiles profile-name]
user@host# edit services captive-portal-content-delivery
```

3. Create a rule to apply to traffic destined outside the walled garden.

```
[edit dynamic-profiles profile-name services captive-portal-content-delivery]
edit rule name
```

4. Specify that the rule applies to incoming traffic.

```
[edit dynamic-profiles profile-name services captive-portal-content-delivery rule name]
user@host# set match-direction input
```

5. Specify the action to take for the matching traffic. Because the walled garden is a service filter, the traffic is already identified as HTTP traffic before being sent to the service.
   - For a local HTTP redirect server, provide the redirect URL, which is the URL of the captive portal with the original URL (outside the walled garden) appended:

```
[edit dynamic-profiles profile-name services captive-portal-content-delivery rule name]
user@host# set term name then redirect redirect-url/url=%dest-url%
```

- For a remote HTTP redirect server, provide the destination address of the remote server:

```
[edit dynamic-profiles profile-name services captive-portal-content-delivery rule name]
user@host# set term name then rewrite destination-address remote-server-address
```

For example, in the following configuration for a local server, the dynamic profile http-redir-converged includes the CPCD service rule redir-svc. The rule redirects traffic to a captive portal, `http://www.portal.example.com`. The original URL entered by the subscriber is appended to the redirect URL. The CPCD service profile redir-prof includes the rule, and will later be applied to a service interface by a service set.

```
user@host# edit dynamic-profiles http-redir-converged
user@host# edit services captive-portal-content-delivery
user@host# edit rule redir-svc
user@host# set match-direction input
user@host# set term redir1 then redirect http://www.portal.example.com/url=%dest-url%
```

The following configuration for a remote server creates CPCD service rule rewr-svc that rewrites the original destination address to the address of the remote server, 192.0.2.230.

```
user@host# edit dynamic-profiles http-redir-converged
user@host# edit services captive-portal-content-delivery
user@host# edit rule rewr-svc
user@host# set match-direction input
user@host# set term rewr1 then rewrite destination-address 192.0.2.230
```

Configuring Parameterization for the Redirect URL

You can optionally choose to parameterize the redirect URL and the rewrite destination address by specifying user-defined variables in the dynamic profile. Parameterizing means that URL or address becomes a dynamic variable. The value is provided by RADIUS when the subscriber is authenticated or when a CoA is received. Consequently, you can use the RADIUS attributes to provide different URLs or destination addresses for different subscribers.

1. Configure the dynamic profile.

```
[edit]
user@host# edit dynamic-profiles profile-name
```

2. Access the custom variable configuration level.

```
[edit dynamic-profiles profile-name]
user@host# edit variables
```

3. Define the variable for the redirect URL, the rewrite destination address, or both. Specify that the value for the dynamic variable is provided by an external server, typically RADIUS.

> **NOTE**: You can name the variables anything you like, but names like redirect-url and rewrite-da make the purpose clear.

```
[edit dynamic-profiles profile-name variables]
set variable-name mandatory
```

4. In the CPCD rule, specify the variable by prepending a dollar sign ($) to the variable name.
   - For a local HTTP redirect server, provide the redirect variable:

     ```
     [edit dynamic-profiles profile-name services captive-portal-content-delivery rule name]
     user@host# set term name then redirect $variable-name
     ```

   - For a remote HTTP redirect server, provide the destination address variable:

     ```
     [edit dynamic-profiles profile-name services captive-portal-content-delivery rule name]
     user@host# set term name then rewrite $variable-name
     ```

For example, the following configuration shows two user-defined variables, redirect-url and rewrite-da that require externally provided values when they are instantiated. CPCD service rule redir1 specifies traffic is redirected to $redirect-url. CPCD service rule rewr1 specifies that the destination address for the traffic is rewritten to $rewrite-da.

```
user@host# edit dynamic-profiles http-redir-converged
user@host# edit variables
user@host# set redirect-url mandatory
user@host# set rewrite-da mandatory
user@host# edit services captive-portal-content-delivery
user@host# edit rule redir-svc
user@host# set match-direction input
```

```
user@host# set term redir1 then redirect $redirect-url
user@host# edit rule rewr-svc
user@host# set match-direction input
user@host# set term rewr1 then rewrite $rewrite-da
```

Configuring the Service Set to Associate the Service Profile with a Service Interface

Service sets define one or more services to be performed by the Routing Engine. For HTTP redirect services, you define a CPCD service profile that includes CPCD rules. The service set applies the CPCD service profile to a specific service interface.

1. Create the service profile.

```
[edit services captive-portal-content-delivery]
user@host# edit profile name
```

2. Specify one or more CPCD rules configured in the CPCD dynamic profile for the service profile.

```
[edit services captive-portal-content-delivery profile name]
user@host# set cpcd-rules rule-name
```

3. Specify that this is a converged CPCD service.

```
[edit services captive-portal-content-delivery profile name]
user@host# set dynamic
```

4. Create the service set.

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

5. Specify that the service set is for Routing Engine–Based CPCD.

```
[edit services service-set name]
user@host# set service-set-options routing-engine-services
```

6. Specify the CPCD service profile.

```
[edit services service-set name]
user@host# set captive-portal-content-delivery-profile name
```

7. Specify the service interface.

```
[edit services service-set name]
user@host# set interface-service service-interface interface-name
```

For example, the following configuration creates CPCD service profile redir-prof, which references the CPCD rule redir-svc. Service set cvgd associates the CPCD service profile rewr-prof with the service interface si-4/0/0.

```
[edit services captive-portal-content-delivery]
user@host# edit profile redir-prof
user@host# set cpcd-rules redir-svc
user@host# set dynamic
[edit services]
user@host# edit service-set cvgd
user@host# set captive-portal-content-delivery-profile redir-prof
user@host# set interface-service service-interface si-4/0/0
```

Attaching a CPCD Service Set and Service Filter to a Dynamic Logical Interface

To use the HTTP redirect services, you must attach the CPCD service set to a logical interface. Because the walled garden is configured as a service filter, you must attach it to the same interface as the service set. Traffic arriving on and leaving that interface is filtered by the service filter. Traffic identified for servicing is sent to the Routing Engine service interface where the CPCD profile is applied.

NOTE: This procedure shows only elements of the dynamic profile configuration that are specific to the converged services configuration. The complete dynamic profile depends on your use case.

1. Configure the dynamic profile.

```
[edit]
user@host# edit dynamic-profiles profile-name
```

2. Configure the dynamic physical interface.

```
[edit dynamic-profiles profile-name]
user@host# edit interfaces $junos-interface-ifd-name
```

3. Configure the dynamic logical interface.

```
[edit dynamic-profiles profile-name interfaces $junos-interface-ifd-name]
user@host# edit unit $junos-underlying-interface-unit
```

4. Configure the address family.

```
[edit dynamic-profiles profile-name interfaces $junos-interface-ifd-name unit $junos-
underlying-interface-unit]
user@host# edit family family
```

5. Attach the service set and service filter to the interface.

```
[edit dynamic-profiles profile-name interfaces $junos-interface-ifd-name unit $junos-
underlying-interface-unit family family]
user@host# set service input service-set set-name service-filter filter-name
user@host# set service output service-set set-name service-filter filter-name
```

For example, the following configuration creates the dynamic profile http-redir-converged. It specifies predefined variables to create the dynamic physical and logical interfaces in the IPv4 address family. The profile attaches service set cvgd and service filter walled-v4 to the dynamic logical interface when it is created at subscriber login. The service set and filter are both applied to the interface input and output.

```
user@host# edit dynamic-profiles http-redir-converged
user@host# edit interfaces $junos-interface-ifd-name
user@host# edit unit $junos-underlying-interface-unit
user@host# edit family inet
user@host# set service input service-set cvgd service-filter walled-v4
user@host# set service output service-set cvgd service-filter walled-v4
```

RELATED DOCUMENTATION

Dynamic Profiles Overview

Dynamic Variables Overview

Junos OS Predefined Variables

User-Defined Variables

HTTP Redirect Service Overview | **511**

Remote HTTP Redirect Server Operation Flow | **518**

Local HTTP Redirect Server Operation Flow (MX Series, ACX7100-48L, ACX7332 and ACX7348) | **520**

Configuring Routing Engine-Based, Static HTTP Redirect Services | **544**

Configuring MS-MPC-Based or MX-SPC3-Based Converged HTTP Redirect Services | **532**

Adding Subscriber Information to HTTP Redirect URLs | **570**

Adding Subscriber Information to HTTP Redirect URLs

NOTE: Starting in Junos OS Release 19.3R2, the HTTP redirect service is also supported if you have enabled Next Gen Services on the MX Series.

Starting in Junos OS Release 17.3R1, you can add subscriber information to a redirect URL to make it easier to track subscribers, change service policies, and provision services. For example, a WLAN service model might redirect subscribers to a captive portal when they connect to the network and open a browser. The captive portal may provide an opportunity to update or purchase new services or require subscribers to enter their credentials before they can access a service. For example, the subscriber might be offered an opportunity to pay for a faster Internet connection.

You can configure the Juniper Networks RADIUS VSAs Activate-Service (26-65) or Deactivate-Service (26-66) to specify a format for the redirect URL that includes tokens for several subscriber attributes. The values for these tokens are retrieved from the subscriber session database and appended to the redirect URL. When the CPCD service is activated, the modified redirect URL is then returned to the requesting HTTP client in a message with an HTTP 302 or 307 status code. You can specify the tokens in any order. When the CPCD service is deactivated, the subscriber traffic is no longer redirected; the deactivation effectively removes the redirect rule for the subscriber,

When the subscriber subsequently logs in at the captive portal or purchases new services or updates, the web server hosting the captive portal confirms the action based on the supplied credentials. The server then contacts the RADIUS service to update the service policies for that particular subscriber. The subscriber attributes appended to the redirect URL enable RADIUS to determine exactly which subscriber to update. RADIUS then sends a CoA to the router to update the subscriber's policy and access.

Table 39 on page 571 describes the supported subscriber tokens. If other tokens are included in the redirect URL format in the VSA, they are ignored.

Table 39: Supported subscriber Tokens for Redirect URLs

Token for URL Format	Subscriber Attribute
%subsc-ip%	Subscriber's private IP address.
%subsc-ipv6%	Subscriber's complete private IPv6 address (not just the prefix).
%nas-ip%	BNG IP address, configured with the `router-id` statement at the `[edit routing-options]` hierarchy level.
%ac-name%	This token is always empty on a BNG.
%dest-url%	Original, requested URL.
%nas-port-id%	Subscriber's interface information, contained in the RADIUS NAS-Port-Id attribute (87). The attribute must include the interface name (physical or logical) and the PVLAN or CVLAN identifiers. The VLAN identifiers are in the range 1 through 4095.
%mac-sa%	MAC address of the WLAN client (the device the subscriber uses to access the network).
%sess-id%	Subscriber session ID.
%user-name%	Subscriber username.

NOTE: Refer to your RADIUS server documentation for information about configuring the service VSAs.

Configure the redirect URL with the desired tokens. In the following example, the redirect URL is `http://portal.wifi.example.com`. The tokens are delimited by the & (ampersand) character.

```
http://portal.wifi.example.com/xx?wlanuseraddr=%subsc-ip% &nasaddr=%nas-ip%&url=%dest-url%&userlocation=%nas-port-id% &usermac=%mac-sa%&acname=%ac-name%&session-id=%sess-id% &username=%user-name%
```

The RADIUS service VSA includes the redirect URL with appended tokens in parentheses immediately following the name of the service to be activated—the dynamic service profile. In the following example, the profile is http-redirect-converged2:

```
http-redirect-converged2(http://portal.wifi.example.com/xx?wlanuseraddr=%subsc-ip% &nasaddr=%nas-ip%&url=%dest-url%&userlocation=%nas-port-id% &usermac=%mac-sa%&acname=%ac-name%&session-id=%sess-id% &username=%user-name%
```

As an example, the returned redirect URL might look like the following when the tokens are replaced with the actual subscriber values retrieved from the session database:

```
http://portal.wifi.example.com?wlanuseraddr=192.0.2.66&nasaddr=203.0.113.1 &url=http%3A%2F%2F192.0.2.1%3A80%2Ftest.html&ip=192.0.2.1:80 &userlocation=ge-1/0/0:100&usermac=00:00:5E:00:53:42&acname=&session-id=886&username=USER1@EXAMPLE.NET
```

You can configure adding subscriber information to the redirect URL for dynamic (converged) Routing Engine-based and dynamic MS-MPC/MS-MIC-based or MX-SPC3 services card-based CPCD.

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
17.3R1	Starting in Junos OS Release 17.3R1, you can add subscriber information to a redirect URL to make it easier to track subscribers, change service policies, and provision services.

RELATED DOCUMENTATION

HTTP Redirect Service Overview | 511

How to Automatically Remove the HTTP Redirect Service After the Initial Redirect

In some deployments, you might want to always redirect your subscribers to the captive portal just once each session so that you can serve them advertisements or notifications. Thereafter, you want the subscribers to reach the URL that they specify without additional redirects.

In other deployments, HTTP redirect services might be set up so that the subscriber is redirected multiple times before being able to access the requested URL. For example, after logging in and

requesting a URL, the subscriber is redirected to a payment page. After satisfying the payment requirements, the subscriber again requests the URL, but is redirected to an advertisement page, such as for more service offerings. The subscriber must request the URL again to reach the target. In this business case, you might want to simplify access for certain customers by removing the redirect service after the first redirect.

Removal of the redirect service typically requires action by the external policy server such as the PCRF or RADIUS server. For example, the RADIUS server might send a CoA message to deactivate the service. Starting in Junos OS Release 19.4R1, you can configure the router to automatically remove the redirect service when triggered. You can use this automatic method when you do not want the external policy server to be involved in removing the service. The trigger for automatic removal is the initial HTTP GET request from the subscriber. When the subscriber initially requests the URL, the subscriber is redirected once to the captive portal the first time the URL is requested. That Get request causes the router to remove the redirect service, so that the next request for the URL takes the subscriber directly to that location.

Use one of the following methods to configure the automatic removal feature:

- Enable automatic removal for static redirect services.

```
[edit services captive-portal-content-delivery]
user@host# set auto-deactivate initial-get
```

- Disable automatic removal for static redirect services.

```
[edit services captive-portal-content-delivery]
user@host# set auto-deactivate never
```

- Enable automatic removal for dynamic redirect services.

```
[edit dynamic-profiles profile-name services captive-portal-content-delivery]
user@host# set auto-deactivate initial-get
```

- Disable automatic removal for dynamic redirect services.

```
[edit dynamic-profiles profile-name services captive-portal-content-delivery]
user@host# set auto-deactivate never
```

For dynamic HTTP redirect services, you can also create a user-defined variable to enable or disable automatic removal. The variable value, `initial-get` or `never`, is supplied by either the external policy server or a default value that you define. To use the variable:

1. Specify the user-defined variable in the dynamic profile.

```
[edit dynamic-profiles profile-name services captive-portal-content-delivery]
user@host# set auto-deactivate $variable-name
```

2. Configure your external policy server to provide the value. See your server documentation for information about how to do this.
3. (Optional) Define a default value for the variable.

```
[edit dynamic-profiles profile-name]
user@host# set variables variable-name default-value default-value
```

For example, the following configuration specifies that in the absence of information from the external server, the initial GET message triggers automatic removal of the redirect service.

```
[edit dynamic-profiles profile-name services]
user@host# set captive-portal-content-delivery auto-deactivate $remove-redirect-service
user@host# set variables remove-redirect-service default-value initial-get
```

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
19.4R1	Starting in Junos OS Release 19.4R1, you can configure the router to automatically remove the redirect service when triggered

RELATED DOCUMENTATION

- HTTP Redirect Service Overview | 511
- Configuring MS-MPC-Based or MX-SPC3-Based Static HTTP Redirect Services | 522
- Configuring MS-MPC-Based or MX-SPC3-Based Converged HTTP Redirect Services | 532
- Configuring Routing Engine-Based, Static HTTP Redirect Services | 544

Example: Configuring HTTP Redirect Services Using a Next-Hop Method and Attaching It to a Static Interface

IN THIS SECTION

- Requirements | 575
- Overview | 575
- Configuration | 576
- Verification | 592

This example shows how to configure HTTP redirect services using a next-hop method and attaching it to a static interface.

Requirements

This example uses the following hardware and software components:

- MX240, MX480, or MX960 Universal Routing Platform with a Multiservices Modular PIC Concentrator (MS-MPC) and Multiservices Modular Interfaces Card (MS-MIC) installed.
- Junos OS Release 15.1 or later.

Before you begin:

- Configure the connection between the redirect server and the MX Series router.
- Define the source address (203.0.113.0/24 is used in this example).
- Define one or more interfaces used for subscriber traffic.

Overview

HTTP redirect and rewrite services are supported for both IPv4 and IPv6. You can attach an HTTP redirect service or service set to either a static or dynamic interface. For dynamic subscriber management, you can attach HTTP services or service sets dynamically at subscriber login or by using a change of authorization (CoA). Using a next-hop method, you can configure HTTP redirect services and attach it to a static interface.

Configuration

IN THIS SECTION

- CLI Quick Configuration | **576**
- Configuring the CPCD Services and Attaching Service Set to Static Interface | **578**
- Configuring the Package and Installation for CPCD | **580**
- Configuring the Static Interface, HTTP Redirect Filters, and Interface Service Options | **581**
- Configuring the Additional Routing Instance and Assigning Its Next-Hop Static Interfaces | **585**
- Configuring the Interface-Specific Filters to Direct HTTP Traffic | **587**
- Configuring the Policy Option and Statement to Use a Private Blocks Prefix List | **590**
- Using Broadband Edge Static Route Configuration for Subscriber (Junos OS Release 23.4R1 for MX-Series Devices) | **591**

To configure HTTP redirect services using a next-hop method and attach it to a static interface, 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, and then copy and paste the commands into the CLI.

```
[edit]
edit services captive-portal-content-delivery
set rule redirect match-direction input
set rule redirect term REDIRECT then redirect http://redirection-portal/redirection/
set profile http-redirect cpcd-rules redirect
edit services service-set http-redirect-sset
set captive-portal-content-delivery-profile http-redirect
set next-hop-service inside-service-interface ms-11/1/0.1
set next-hop-service outside-service-interface ms-11/1/0.2

[edit]
edit chassis fpc 11 pic 1 adaptive-services service-package
set extension-provider package jservices-cpcd
set extension-provider syslog daemon none
set extension-provider syslog external none
```

```
set extension-provider syslog kernel none
set extension-provider syslog pfe none

[edit]
set interfaces ge-0/0/1 unit 900 description VLAN REDIRECT
set interfaces ge-0/0/1 unit 900 vlan-id 900
set interfaces ge-0/0/1 unit 900 family inet filter input FF_HTTP_REDIR_IN
set interfaces ge-0/0/1 unit 900 family inet address 203.0.113.250/30
edit interfaces ms-11/1/0 services-options open-timeout 4
edit interfaces ms-11/1/0 services-options close-timeout 2
edit interfaces ms-11/1/0 services-options inactivity-tcp-timeout 5
edit interfaces ms-11/1/0 services-options inactivity-non-tcp-timeout 5
edit interfaces ms-11/1/0 services-options session-timeout 5
edit interfaces ms-11/1/0 services-options tcp-tickles 0
set interfaces ms-11/1/0 unit 1 family inet
set interfaces ms-11/1/0 unit 1 service-domain inside
set interfaces ms-11/1/0 unit 2 filter output FF_CPCD_REDIRECT_OUTPUT
set interfaces ms-11/1/0 unit 2 family inet
set interfaces ms-11/1/0 unit 2 service-domain outside

[edit]
edit routing-instances CPCD_REDIRECT
set instance-type virtual-router
set interface ms-1/1/0.1
set interface ms-1/1/0.2
set routing-options static route 0.0.0.0/0 next-hop ms-1/1/0.1
set routing-options static route 203.0.113.0/24 next-hop ms-1/1/0.2

[edit]
edit firewall family inet
set filter FF_CPCD_REDIRECT_OUTPUT interface-specific
set filter FF_CPCD_REDIRECT_OUTPUT term One then count back-to-default
set filter FF_CPCD_REDIRECT_OUTPUT term One then routing-instance default
set filter FF_HTTP_REDIR_IN interface-specific
set filter FF_HTTP_REDIR_IN term ACCEPTED_PREFIXES from prefix-list User-PRIVATE-Blocks-01
set filter FF_HTTP_REDIR_IN term ACCEPTED_PREFIXES then next term
set filter FF_HTTP_REDIR_IN term HTTP from protocol tcp
set filter FF_HTTP_REDIR_IN term HTTP from destination-port http
set filter FF_HTTP_REDIR_IN term HTTP then count HTTP
set filter FF_HTTP_REDIR_IN term HTTP then forwarding-class best-effort
set filter FF_HTTP_REDIR_IN term HTTP then routing-instance CPCD_REDIRECT

[edit]
```

```
edit policy-options policy-statement User-PRIVATE-Blocks-01
set 203.0.113.0/24
```

Configuring the CPCD Services and Attaching Service Set to Static Interface

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

1. Configure the HTTP redirect service by specifying the location to which a subscriber's initial Web browser session is redirected, enabling initial provisioning and service selection for the subscriber.

```
[edit services]
user@host# edit captive-portal-content-delivery
```

2. Configure the service filter as a walled garden by defining the rule the router references when applying this HTTP service.

```
[edit services captive-portal-content-delivery]
user@host# edit rule redirect
```

3. Specify that the rule matches traffic coming in on the interface.

```
[edit services captive-portal-content-delivery rule redirect]
user@host# match-direction input
```

4. Create the term match and action properties for the CPCD rule for the HTTP service.

```
[edit services captive-portal-content-delivery rule redirect]
user@host# set term REDIRECT then redirect http://redirection-portal/redirection/
```

5. Create the CPCD profile for the IP destination address to redirect the HTTP service.

```
[edit services captive-portal-content-delivery]
user@host# edit profile http-redirect
```

6. Specify the CPCD rule for the HTTP service.

```
[edit services captive-portal-content-delivery profile http-redirect]
user@host# set cpcd-rules redirect
```

7. Create the service set for the CPCD services.

```
[edit services service-set]
user@host# edit http-redirect-sset
```

8. Specify the CPCD profile for the service set.

```
[edit services service-set http-redirect-sset]
user@host# set captive-portal-content-delivery-profile http-redirect
```

9. Specify the interface name for the next-hop service for an inside and outside service interfaces and attach them to static interfaces.

```
[edit services service-set http-redirect-sset]
user@host# set next-hop-service inside-service-interface ms-11/1/0.1
user@host# set next-hop-service outside-service-interface ms-11/1/0.2
```

Results

From configuration mode, confirm your configuration by entering the `show services` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
root@host# show services
captive-portal-content-delivery {
    rule redirect {
        match-direction input;
        term REDIRECT {
            then {
                redirect http://redirection-portal/redirection/;
            }
        }
```

```
    }
    profile http-redirect {
        cpcd-rules redirect;
    }
}
service-set http-redirect-sset {
    captive-portal-content-delivery-profile http-redirect;
    next-hop-service {
        inside-service-interface ms-11/1/0.1;
        outside-service-interface ms-11/1/0.2;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring the Package and Installation for CPCD

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

1. Configure Junos OS to support the service package on a service interface on an MX Series 5G Universal Routing Platform with MS-MPCs/MS-MICs.

```
[edit chassis]
user@host# edit fpc 11 pic 1 adaptive-services service-package
```

2. Configure the CPCD service package to run on the PIC. When the `extension-provider` statement is first configured, the PIC reboots.

```
[edit chassis fpc 11 pic 1 adaptive-services service-package]
user@host# set extension-provider package jservices-cpcd
```

3. Enable PIC system logging to record or view system log messages on the PIC but do not include daemon, external, kernel, or Packet Forwarding Engine processes.

```
[edit chassis fpc 11 pic 1 adaptive-services service-package extension-provider]
user@host# set extension-provider syslog daemon none
user@host# set extension-provider syslog external none
```

```
user@host# set extension-provider syslog kernel none
user@host# set extension-provider syslog pfe none
```

Results

From configuration mode, confirm your configuration by entering the `show chassis` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
root@host# show chassis
    fpc 11 {
        pic 1 {
            adaptive-services {
                service-package {
                    extension-provider {
                        package jservices-cpcd;
                        syslog {
                            daemon none;
                            external none;
                            kernel none;
                            pfe none;
                        }
                    }
                }
            }
        }
    }
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring the Static Interface, HTTP Redirect Filters, and Interface Service Options

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

1. Configure a Gigabit interface with a logical interface on which traffic arrives before it is redirected.

```
[edit interfaces]
user@host# edit ge-0/0/1 unit 900
```

2. Assign a description and VLAN ID to the logical interface.

```
[edit interfaces ge-0/0/1 unit 900]
user@host# set description VLAN-REDIRECT
user@host# set vlan-id 900
```

3. Configure the IPv4 family for the interface.

```
[edit interfaces ge-0/0/1 unit 900]
user@host# edit family inet
```

4. Configure an input filter to evaluate when packets are received and redirected on the interface.

```
[edit interfaces ge-0/0/1 unit 900 family inet]
user@host# set filter input FF_HTTP_REDIR_IN
```

5. Configure an address for the input filter.

```
[edit interfaces ge-0/0/1 unit 900 family inet]
user@host# set address 203.0.113.250/30
```

6. Configure service options to be applied on the Multiservices interface.

```
[edit interfaces]
user@host# edit ms-11/1/0 services-options
```

NOTE: The values configured for the service options are shown for example only. You must configure and provision appropriate values as per the requirement.

7. Specify the open and close timeout periods in seconds for Transmission Control Protocol (TCP) session establishment.

```
[edit interfaces ms-11/1/0 services-options]
user@host# set open-timeout 4
user@host# set close-timeout 2
```

8. Specify the inactivity timeout periods in seconds for established TCP and non-TCP sessions.

```
[edit interfaces ms-11/1/0 services-options]
user@host# set inactivity-tcp-timeout 5
set inactivity-non-tcp-timeout 5
```

9. Specify the session lifetime in seconds globally for the Multiservices interface.

```
[edit interfaces ms-11/1/0 services-options]
user@host# set session-timeout 5
```

10. Specify the maximum number of keep-alive messages sent before a TCP session is allowed to time out.

```
[edit interfaces ms-11/1/0 services-options]
user@host# set tcp-tickles 0
```

11. Configure a logical interface on the Multiservices interface.

```
[edit interfaces ms-11/1/0]
user@host# edit unit 1
```

12. Configure the service domain to specify that the logical interface is used within the network.

```
[edit interfaces ms-11/1/0 unit 1]
user@host# set service-domain inside
```

13. Configure the IPv4 address family on the logical interface.

```
[edit interfaces ms-11/1/0 unit 1]
user@host# set family inet
```

14. Configure a second logical interface on the Multiservices interface.

```
[edit interfaces ms-11/1/0]
user@host# edit unit 2
```

15. Configure the service domain to specify that the logical interface is used outside the network.

```
[edit interfaces ms-11/1/0 unit 2]
user@host# set service-domain outside
```

16. Configure an output filter to redirect CPCD packets from the logical interface.

```
[edit interfaces ms-11/1/0 unit 2]
user@host# set filter output FF_CPCD_REDIRECT_OUTPUT
```

17. Configure the IPv4 address family on the logical interface.

```
[edit interfaces ms-11/1/0 unit 2]
user@host# set family inet
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
root@host# show interfaces
ge-0/0/1 {
    unit 900 {
        description VLAN-REDIRECT;
        vlan-id 900;
```

```
    }
    family inet {
        filter {
            input FF_HTTP_REDIR_IN;
        }
        address 203.0.113.250/30;
    }
}
ms-11/1/0 {
    services-options {
        open-timeout 4;
        close-timeout 2;
        inactivity-tcp-timeout 5;
        inactivity-non-tcp-timeout 5;
        session-timeout 5;
        tcp-tickles 0;
    }
    unit 1 {
        family inet;
        service-domain inside;
    }
    unit 2 {
        family inet {
            filter {
                output FF_CPCD_REDIRECT_OUTPUT;
            }
        }
        service-domain outside;
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring the Additional Routing Instance and Assigning Its Next-Hop Static Interfaces

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

1. Configure a routing instance.

```
[edit routing-instances]
user@host# edit CPCD_REDIRECT
```

2. Configure a virtual router routing instance.

```
[edit routing-instances CPCD_REDIRECT]
user@host# set instance-type virtual-router
```

3. Configure the two previously defined multiservices interfaces for the routing instance.

```
[edit routing-instances CPCD_REDIRECT]
user@host# set interface ms-11/1/0.1
user@host# set interface ms-11/1/0.2
```

4. Configure static routing options.

```
[edit routing-instances CPCD_REDIRECT]
user@host# edit routing-options static
```

5. Assign the next-hop static interfaces to the routes and routing instance.

```
[edit routing-instances CPCD_REDIRECT routing-options static]
user@host# set route 0.0.0.0/0 next-hop ms-11/1/0.1
user@host# set route 203.0.113.0/24 next-hop ms-11/1/0.2
```

Results

From configuration mode, confirm your configuration by entering the `show routing-instances` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
root@host# show routing-instances
CPCD_REDIRECT {
    instance-type virtual-router;
```

```
    interface ms-11/1/0.1;
    interface ms-11/1/0.2;
    routing-options {
        static {
            route 0.0.0.0/0 next-hop ms-11/1/0.1;
            route 203.0.113.0/24 next-hop ms-11/1/0.2;
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring the Interface-Specific Filters to Direct HTTP Traffic

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

1. Create a family for the service filter under the `[edit firewall]` hierarchy.

```
[edit firewall]
user@host# edit family inet
```

2. Create an interface-specific filter to redirect output traffic for CPCD.

```
[edit firewall family inet]
user@host# edit filter FF_CPCD_REDIRECT_OUTPUT
```

3. Specify that this is an interface-specific filter.

```
[edit firewall family inet filter FF_CPCD_REDIRECT_OUTPUT]
user@host# set interface-specific
```

4. Create a filter term for the interface-specific filter for the walled garden.

```
[edit firewall family inet filter FF_CPCD_REDIRECT_OUTPUT]
user@host# edit term One
```

5. Specify both the action to count default traffic and the default routing instance.

```
[edit firewall family inet filter FF_CPCD_REDIRECT_OUTPUT interface-specific term One]
user@host# set then count back-to-default
set then routing-instance default
```

6. Create a filter to redirect HTTP input traffic.

```
[edit firewall family inet]
user@host# edit filter FF_HTTP_REDIR_IN
```

7. Specify that this is an interface-specific filter.

```
[edit firewall family inet filter FF_HTTP_REDIR_IN]
user@host# set interface-specific
```

8. Create a filter term for the interface-specific filter for the walled garden.

```
[edit firewall family inet filter FF_HTTP_REDIR_IN]
user@host# edit term ACCEPTED_PREFIXES
```

9. Specify the list of accepted prefixes as a match conditions for the walled garden's filter.

```
[edit firewall family inet filter FF_HTTP_REDIR_IN term ACCEPTED_PREFIXES]
user@host# set from prefix-list User-PRIVATE-Blocks-01
```

10. Specify the action to take for all the matching HTTP traffic.

```
[edit firewall family inet filter FF_HTTP_REDIR_IN term ACCEPTED_PREFIXES]
user@host# set then next term
```

11. Create a second filter term for the walled garden's filter.

```
[edit firewall family inet filter FF_HTTP_REDIR_IN interface-specific]
user@host# edit term HTTP
```

12. Specify the protocol and destination port as match conditions for the walled garden's filter.

```
[edit firewall family inet filter FF_HTTP_REDIR_IN term HTTP]
user@host# set from protocol tcp
user@host# set from destination-port http
```

13. Specify the action to take for matching HTTP traffic destined to flow outside of the walled garden.

```
[edit firewall family inet filter filter FF_HTTP_REDIR_IN interface-specific term HTTP]
user@host# set then count HTTP
user@host# set then forwarding-class best-effort
user@host# set then routing-instance CPCD_REDIRECT
```

Results

From configuration mode, confirm your configuration by entering the `show firewall` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
root@host# show firewall
family inet {
    filter FF_CPCD_REDIRECT_OUTPUT {
        interface-specific;
        term One {
            then {
                count back-to-default;
                routing-instance default;
            }
        }
    }
    filter FF_HTTP_REDIR_IN {
        interface-specific;
        term ACCEPTED_PREFIXES {
            from {
                prefix-list {
                    User-PRIVATE-Blocks-01;
                }
            }
            then next term;
```

```
        }
        term HTTP {
            from {
                protocol tcp;
                destination-port http;
            }
            then {
                count http;
                forwarding-class best-effort;
                routing-instance CPCD_REDIRECT;
            }
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Configuring the Policy Option and Statement to Use a Private Blocks Prefix List

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

1. Create a policy option and statement to use a private blocks prefix list under the `[edit policy-options]` hierarchy.

```
[edit policy-options]
user@host# set policy-statement User-PRIVATE-Blocks-01
```

2. Configure the source address for the private blocks prefix list.

```
[edit policy-options policy-statement User-PRIVATE-Blocks-01]
user@host# set 203.0.113.0/24
```

Results

From configuration mode, confirm your configuration by entering the `show policy-options` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
root@host# show policy-options
policy-statement User-PRIVATE-Blocks-01 {
    203.0.113.0/24;
}
```

If you are done configuring the device, enter `commit` from configuration mode.

Using Broadband Edge Static Route Configuration for Subscriber (Junos OS Release 23.4R1 for MX-Series Devices)

Starting Junos 23.4R1 the broadband edge static route configuration for subscribers feature for BNG replaces the RADIUS framed route configuration. You can now setup static IP addresses for multiple hosts on the same site.

For example:

- Use a pre-existing configuration to add the routes to the routing table. Once this configuration is committed, the routes are hidden until the subscriber with configured subscriber IP comes up.

```
staticRoute
   {
        routing-options
            {
                 access
                    {
                           route 7.7.7.7/32 next-hop 50.1.1.1;
                    }
            }
   }
```

- You can enable static framed-routes feature on the BNG towards a specific customer connection, using the command `static-framed-route` under the `[edit system services subscriber-management]` mode.

```
user@root> set system services subscriber-management static-framed-route
```

- You can now use RADIUS server for authentication purposes and not for sending framed-routes.

> **NOTE**: Static framed routes must be added, modified or deleted only when subscribers are down.
>
> When subscribers come up, static framed-routes are attached to them. Static framed-route is supported for IPv4 only.

Verification

IN THIS SECTION

- Verifying the Configured Service Set for CPCD Services | **592**
- Verifying Details for a Configured HTTP Service Rule for a Walled Garden | **593**

To confirm that HTTP redirect services has been configured correctly within a service set, perform these tasks:

Verifying the Configured Service Set for CPCD Services

Purpose

Display the configured CPCD service set.

Action

From operational mode, enter the `show services captive-portal-content-delivery service-set http-redirect-sset detail` command.

```
user@host> show services captive-portal-content-delivery service-set http-redirect-sset detail
Service Set             Id       Profile         Compiled Rules
```

```
http-redirect-sset       1        http-redirect     1
```

Meaning

The output lists the service set configured for CPCD services.

Verifying Details for a Configured HTTP Service Rule for a Walled Garden

Purpose

Display details for a specific configured HTTP service rule for a walled garden.

Action

From operational mode, enter the `show services captive-portal-content-delivery rule redirect term REDIRECT` command.

```
user@host> show services captive-portal-content-delivery rule redirect term REDIRECT
Rule name: redirect
Rule match direction: input
Term name: term REDIRECT
Term action: redirect
Term action option: http://redirection-portal/redirection/
```

Meaning

The output lists rule and term details for a specific HTTP service rule configured for the walled garden.

RELATED DOCUMENTATION

HTTP Redirect Service Overview | 511

Example: Configuring HTTP Redirect Services Using an Interface-Specific Filter and Attaching It to a Static Interface

PART 7

Configuring Subscriber Secure Policy

- Configuring Subscriber Secure Policy Traffic Mirroring Overview | **595**
- Configuring RADIUS-Initiated Subscriber Secure Policy Traffic Mirroring | **600**
- Configuring DTCP-Initiated Subscriber Secure Policy Traffic Mirroring | **624**
- Configuring DTCP Messages Used for DTCP-Initiated Subscriber Secure Policy Mirroring | **649**
- Configuring Subscriber Secure Policy Support for IPv4 Multicast Traffic | **672**
- Configuring Intercept-Related Information for Subscriber Secure Policy | **675**

CHAPTER 34

Configuring Subscriber Secure Policy Traffic Mirroring Overview

IN THIS CHAPTER

- Subscriber Secure Policy Overview | 595

Subscriber Secure Policy Overview

IN THIS SECTION

- Support for Intercepting Both Layer 2 and Layer 3 Datagrams | 596
- Traffic Filtering for DTCP-Initiated Subscriber Secure Policy Mirrored Traffic | 596
- Mirroring-Related Event Reporting | 596
- Support for L2TP Subscribers | 597
- Junos OS Service for Subscriber Secure Policy Traffic Mirroring | 597
- Protection of SSP Data when a Core Error is Generated | 598
- Subscriber Secure Policy Licensing Requirements | 598

Subscriber secure policy enables you to mirror traffic on a per-subscriber basis. You can mirror the content of subscriber traffic as well as monitor events related to the subscriber session that is being mirrored.

Subscriber secure policy (SSP) mirroring can be based on information provided by either RADIUS or Dynamic Tasking Control Protocol (DTCP), and can mirror both IPv4 and IPv6 traffic. Configuration of subscriber secure policy mirroring is independent of the actual mirroring session—you can configure the mirroring parameters at any time. Also, you can use a single RADIUS or DTCP server to provision mirroring operations on multiple routers in a service provider's network. To provide security, the ability

to configure, access, and view the subscriber secure policy components and configuration is restricted to authorized users.

After subscriber secure policy is triggered, the subscriber's incoming and outgoing traffic are both mirrored. The original traffic is sent to its intended destination and the mirrored traffic is sent to a mediation device for analysis. The actual mirroring operation is transparent to subscribers whose traffic is being mirrored. A special UDP/IP header is prepended to each mirrored packet sent to the mediation device. The mediation device uses the header to differentiate multiple mirrored streams that arrive from different sources.

NOTE: This feature requires a license. To understand more about Subscriber Access Licensing, see, Subscriber Access Licensing Overview. Please refer to the Juniper Licensing Guide for general information about License Management. Please refer to the product Data Sheets at MX Series 5G Universal Routing Platform for details, or contact your Juniper Account Team or Juniper Partner.

Support for Intercepting Both Layer 2 and Layer 3 Datagrams

When DTCP- or RADIUS-initiated SSP intercepts traffic on logical subscriber interfaces and VLAN subscriber interfaces, it sends both Layer 2 and Layer 3 datagrams to the mediation device. When you enable subscriber secure policy for these interfaces, traffic for all configured families (inet, inet6) including Layer 2 and Layer 3 control traffic is mirrored.

Traffic Filtering for DTCP-Initiated Subscriber Secure Policy Mirrored Traffic

You can filter mirrored traffic before it is sent to a mediation device. With this feature, service providers can reduce the volume of traffic sent to a mediation device. For some types of traffic, such as IPTV or video on demand, you do not need to mirror the entire content of the traffic because the content may already be known or controlled by the service provider.

Mirroring-Related Event Reporting

Subscriber secure policy also supports the use of SNMPv3 traps to report events related to the mirroring operation to an external device. Types of information sent in traps include identifying information for subscribers, such as username or IP address, and subscriber session events, such as login or logout events or mirroring session activation or deactivation. The traps map to messages defined in the *Lawfully Authorized Electronic Surveillance (LAES) for IP Network Access, American National Standard for Telecommunications*.

Starting in Junos OS Release 16.1R1, you must configure the target parameters for mediation devices so that the SNMPv3 traps are sent with privacy (encrypted). Targets without privacy configured cannot receive the notifications.

In earlier releases, you can configure target parameters without privacy, allowing unencrypted notifications to be sent to the mediation devices. You also cannot restrict the traps to specific targets.

Support for L2TP Subscribers

Both DTCP-initiated and RADIUS-initiated SSP can be applied to Point-to-Point Protocol (PPP) subscribers whose traffic is tunneled with Layer 2 Tunneling Protocol (L2TP). DTCP-initated SSP and RADIUS-initiated SSP support subscribers at the L2TP access concentrator (LAC) or the L2TP network server (LNS).

At the LAC, both subscriber ingress traffic (from the subscriber into the tunnel) and subscriber egress traffic (from the tunnel to the subscriber) are mirrored at the subscriber-facing ingress interface. The ingress traffic is mirrored after PPPoE decapsulation and before L2TP encapsulation. The egress traffic is mirrored after L2TP decapsulation. The mirrored packet includes the complete HDLC frame sent to the LNS rather than only the IP datagram.

At the LNS, both subscriber ingress traffic (from the LAC to the LNS) and subscriber egress traffic (from the LNS to the LAC) are mirrored at the inline services (si) interface corresponding to the subscriber. Ingress traffic is mirrored after decapsulation of L2TP, HDLC, and PPP headers. The egress traffic is mirrored before the IP datagram is encapsulated. The mirrored traffic contains only the IP datagram belonging to the subscriber.

There is no specific L2TP SSP configuration.

Junos OS Service for Subscriber Secure Policy Traffic Mirroring

Subscriber secure policy mirroring requires the use of the radius-flow-tap service, configured at the `[edit services radius-flow-tap]` hierarchy level. This service is used only for subscriber secure policy mirroring and only on MX Series routers.

There are other Junos OS services with similar names, but they are not used for subscriber secure policy mirroring:

- The flow-tap service, configured at the `[edit services flow-tap]` hierarchy level, is an older Junos OS service for packet mirroring. This service uses Dynamic Tasking Control Protocol (DTCP) requests from mediation devices to intercept IPv4 packets in an active flow monitoring station (router). The router uses DTCP to send a copy of packets that match filter criteria to one or more content destinations. The flow-tap service is supported only on M Series and T Series routers using Adaptive Services PICs. For information about the flow-tap service, see Understanding Flow-Tap Architecture.

- The FlowTapLite service is a lightweight version of the flow-tap service for packet mirroring. It is also configured at the `[edit services flow-tap]` hierarchy level. The FlowTapLite service resides on the Packet Forwarding Engine rather than a line card. The intercepted packets are sent to a tunnel logical interface (vt-) for encapsulation, so you must allocate and assign tunnel interfaces for the service. It is

supported on MX Series routers and on M320 routers with Enhanced III Flexible PIC Concentrators (FPCs). You cannot run FlowTapLite and the flow-tap service on the same router concurrently. For information about FlowTapLite, see Configuring FlowTapLite on MX Series Routers and M320 Routers with FPCs.

Protection of SSP Data when a Core Error is Generated

When the authd, bbe-smgd, or dfcd processes generate a core error, the core dump file contains information related to whatever the process is involved with, including SSP. The error files contain SSP information that might identify the subscriber whose traffic is mirrored or the mediation device that receives the mirrored traffic. For example, the files include information such as the source and destination IP address for the mediation device, device ports, and intercept ID.

Starting in Junos OS Release 18.4R1, SSP-related information is automatically encrypted in core dump files to prevent this information from being seen by unauthorized persons in the event of a core error. Encryption is enabled by default; no configuration is required or possible. The dfcd core error files may contain traffic mirroring information that does not identify subscribers or devices; this information is not masked. FlowTapLite information is not masked.

NOTE: Information related to SSP is not encrypted when it is in a transient state; for example, if the core error occurs when the data has been received from a RADIUS or DTCP server, but is not yet encrypted.

Subscriber Secure Policy Licensing Requirements

To enable and use subscriber secure policy, you must install and properly configure the Subscriber Secure Policy license.

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, SSP-related information is automatically encrypted in core dump files to prevent this information from being seen by unauthorized persons in the event of a core error.

RELATED DOCUMENTATION

RADIUS-Initiated Subscriber Secure Policy Overview | 600

DTCP-Initiated Subscriber Secure Policy Overview | 624

CHAPTER 35

Configuring RADIUS-Initiated Subscriber Secure Policy Traffic Mirroring

IN THIS CHAPTER

- RADIUS-Initiated Subscriber Secure Policy Overview | 600
- Subscriber Secure Policy Traffic Mirroring Architecture Using RADIUS | 601
- RADIUS-Initiated Traffic Mirroring Interfaces | 603
- RADIUS-Initiated Traffic Mirroring Process at Subscriber Login | 605
- RADIUS-Initiated Traffic Mirroring Process for Logged-In Subscribers | 607
- RADIUS Attributes Used for Subscriber Secure Policy | 608
- Using the Packet Header to Track Subscribers on the Mediation Device | 610
- Configuring RADIUS-Initiated Subscriber Secure Policy Mirroring Overview | 616
- Guidelines for Configuring Subscriber Secure Policy Mirroring | 617
- Configuring Support for Subscriber Secure Policy Mirroring | 619
- Configuring RADIUS Server Support for Subscriber Secure Policy Mirroring | 622
- Terminating RADIUS-Initiated Subscriber Traffic Mirroring | 623

RADIUS-Initiated Subscriber Secure Policy Overview

RADIUS-initiated mirroring creates secure policies based on RADIUS VSAs and uses RADIUS attributes to identify the subscriber whose traffic is to be mirrored. Mirroring is initiated without regard to the subscriber location, router, interface, or type of traffic.

Starting Junos OS Release 23.4R1, you can configure static framed-routes towards subscriber in the BNG router itself as an alternative for RADIUS framed routes.

A pre-existing configuration is used to add the routes to the routing table. The routes are hidden until the subscriber with configured subscriber IP comes up.

The mirroring operation can be initiated by RADIUS messages as follows:

- Subscriber login—Mirroring starts when the subscriber logs in and the router receives the trigger in a RADIUS Access-Accept message. Using triggers in RADIUS Access-Accept messages enables you to mirror per-subscriber traffic without regard to how often the subscriber logs in or out, or which router or interface the subscriber uses.
- In-session—Mirroring starts when the router receives the trigger in a RADIUS change of authorization request (CoA-Request) message. Using triggers in CoA-Request messages enables you to immediately mirror traffic of a subscriber who is already logged in.

RELATED DOCUMENTATION

Subscriber Secure Policy Traffic Mirroring Architecture Using RADIUS | 601

Configuring RADIUS-Initiated Subscriber Secure Policy Mirroring Overview | 616

Subscriber Secure Policy Traffic Mirroring Architecture Using RADIUS

Figure 11 on page 601 shows the architecture of the RADIUS-initiated subscriber secure policy mirroring environment.

Figure 11: RADIUS-Initiated Subscriber Secure Policy Architecture

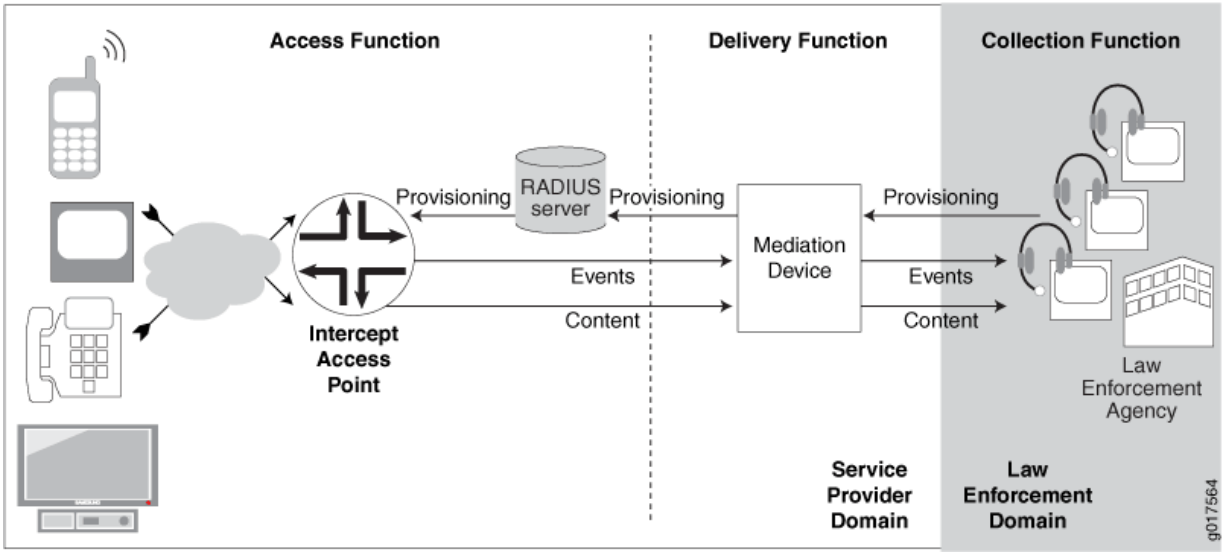


Table 40 on page 602 describes the functions and components of a RADIUS-initiated subscriber secure policy traffic mirroring environment.

Table 40: RADIUS-Initiated Subscriber Secure Policy Functions and Components

Function or Component	Description
Collection function	The collection function is responsible for collecting intercepted content and identifying information from the delivery function. The collection function is the responsibility of the law enforcement agency (LEA).
Delivery function	The delivery function delivers information that it receives from the access function to the collection function. The delivery function is performed by the mediation device.
Access function	The access function has access to the intercept target's traffic content and intercept-related events. It is responsible for collecting this information and sending it to the delivery function. The access function is the responsibility of intercept access points (IAPs).
Events	Intercept-related events, such as login or logout events or mirroring session activation or deactivation. The router sends the events to the mediation device in SNMP traps.
LEA	Law enforcement agency. The LEA provides intercept targets to the service provider who provisions the mediation device.
Mediation device	The mediation device receives provisioning information from the LEA, and it uses the information to send provisioning information to the RADIUS server. The mediation device also receives intercept-related events and intercepted content from the router, and delivers the events and intercepted content to the LEA.

Table 40: RADIUS-Initiated Subscriber Secure Policy Functions and Components *(Continued)*

Function or Component	Description
RADIUS server	The RADIUS server receives provisioning information from the mediation device. It identifies subscribers whose traffic is to be mirrored, and triggers mirroring sessions on the IAP (the router) by including mirroring-related RADIUS attributes and VSAs in Access-Accept or CoA-Request messages that it sends to the IAP.
IAP	Intercept access point. In a subscriber access network the Juniper Networks router is the IAP. Using subscriber secure policies, the IAP intercepts traffic to and from the subscriber whose traffic is being mirrored. It encapsulates the intercepted content in a packet header and delivers it to the mediation device, while also sending the content to the intended destination. The IAP also sends intercept-related events to the mediation device using SNMP traps.

RELATED DOCUMENTATION

- RADIUS-Initiated Subscriber Secure Policy Overview | **600**
- RADIUS-Initiated Traffic Mirroring Interfaces | **603**
- RADIUS-Initiated Traffic Mirroring Process at Subscriber Login | **605**
- RADIUS-Initiated Traffic Mirroring Process for Logged-In Subscribers | **607**

RADIUS-Initiated Traffic Mirroring Interfaces

Figure 12 on page 604 shows the interfaces involved in RADIUS-initiated secure subscriber policy traffic mirroring.

Figure 12: RADIUS-Initiated Traffic Mirroring Interfaces

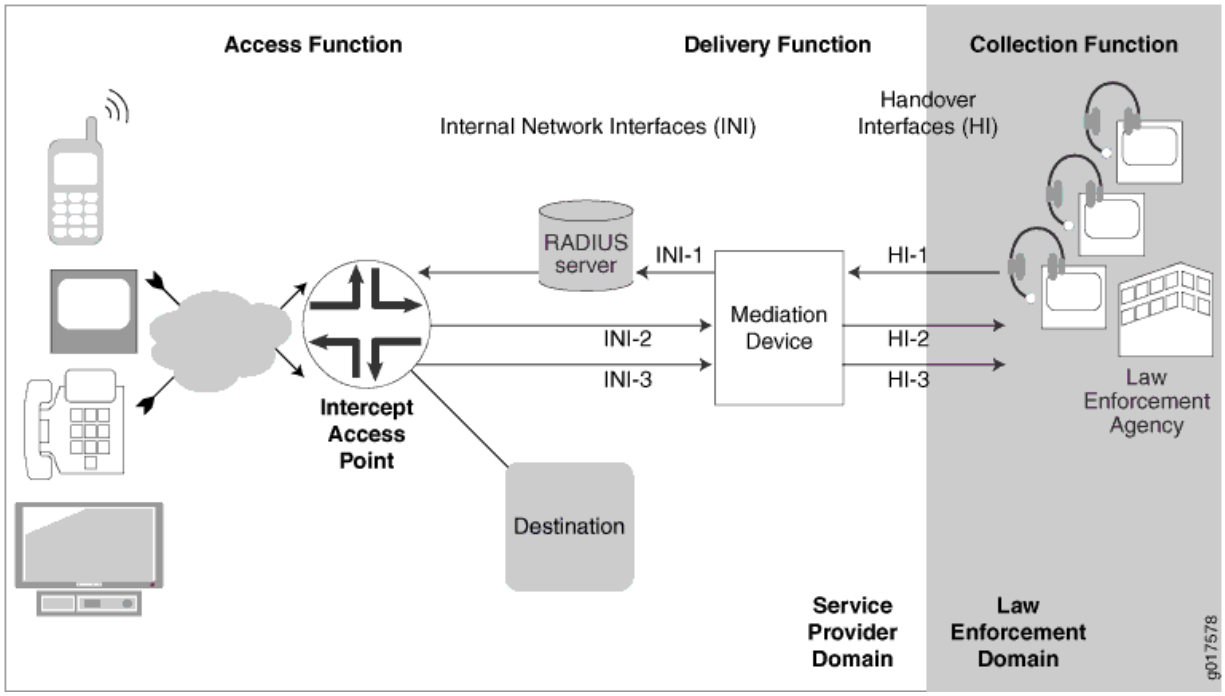


Table 41 on page 604 describes the interfaces involved in RADIUS-initiated secure subscriber policy traffic mirroring.

Table 41: RADIUS-Initiated Traffic Mirroring Interfaces

Interface	Description
HI-1	Handover Interface 1—Administrative interface between the LEA and the service provider mediation device. The LEA sends provisioning information to the mediation device on this interface.
HI-2	Handover Interface 2—Intercept-related information interface between the LEA and the mediation device that is used to deliver intercept-related events to the LEA. These events can be subscriber session events such as login, logout, and authentication.
HI-3	Handover Interface 3—Intercepted content interface between the mediation device and LEA that is used to deliver intercepted content to the LEA.

Table 41: RADIUS-Initiated Traffic Mirroring Interfaces *(Continued)*

Interface	Description
INI-1	Internal network Interface 1—Interface used to send intercept provisioning information from the mediation device to the RADIUS server.
INI-2	Internal network interface 2—Interface used to send intercept-related events from the router to the mediation device. This information is sent in SNMP traps.
INI-3	Internal network interface 3—Interface used to send intercepted content from the router to the mediation device.

RELATED DOCUMENTATION

Subscriber Secure Policy Traffic Mirroring Architecture Using RADIUS | **601**

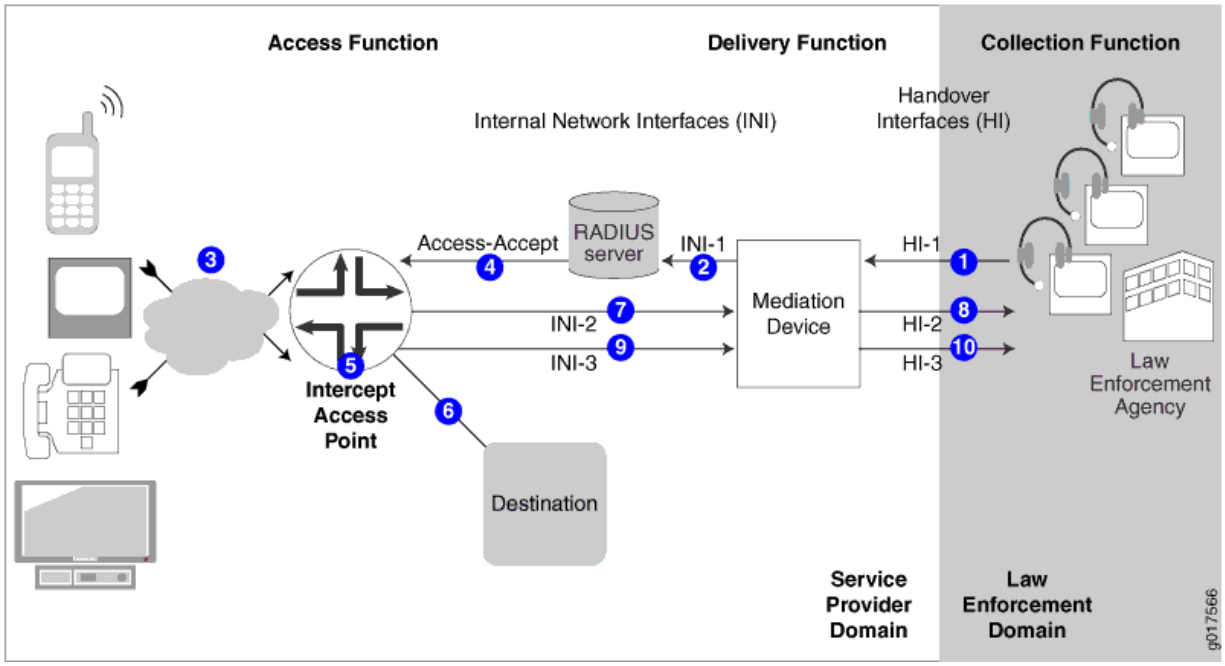
RADIUS-Initiated Traffic Mirroring Process at Subscriber Login | **605**

RADIUS-Initiated Traffic Mirroring Process for Logged-In Subscribers | **607**

RADIUS-Initiated Traffic Mirroring Process at Subscriber Login

Figure 13 on page 606 shows the process for a RADIUS-initiated subscriber mirroring operation that is initiated when the mirrored subscriber logs in.

Figure 13: RADIUS-Initiated Subscriber Secure Policy Model at Login



1– The LEA sends provisioning information for a subscriber whose traffic is to be mirrored over the HI-1 interface to the mediation device.	6– The IAP sends the original subscriber traffic to its intended destination.
2– The mediation device sends the provisioning information over the INI-1 interface to the RADIUS server.	7– As subscriber-related events occur, the IAP sends the events in SNMP traps over the INI-2 interface to the mediation device.
3– The subscriber logs in, requesting authentication by the RADIUS server.	8– The mediation device provides the events over the HI-2 interface to the LEA.
4– The RADIUS server authenticates the subscriber and sends an Access-Accept message containing mirroring-related RADIUS attributes in Juniper Networks VSAs to the IAP (the router).	9– The IAP encapsulates the mirrored content in a packet header and sends it over the INI-3 interface to the mediation device. The IAP uses the destination IP address of the mediation device that it received in the Access-Accept messaged from the RADIUS server.
5– The IAP creates a subscriber secure policy based on the mirroring VSAs and begins mirroring the subscriber's traffic.	10– The mediation device sends mirrored content over the HI-3 interface to the LEA.

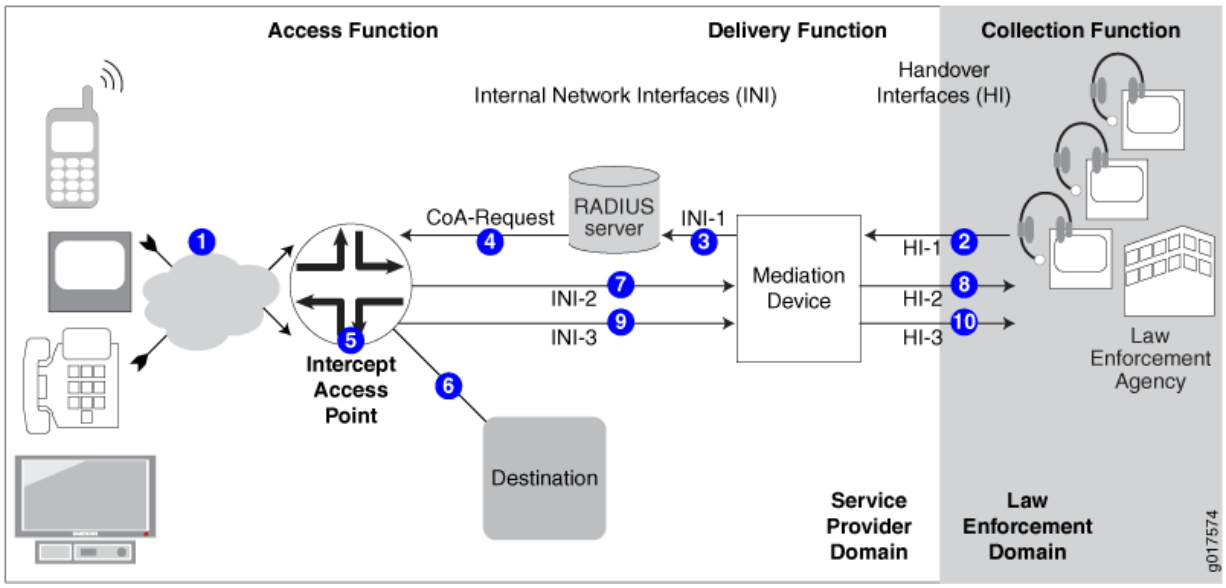
RELATED DOCUMENTATION

- Subscriber Secure Policy Traffic Mirroring Architecture Using RADIUS | 601
- RADIUS-Initiated Traffic Mirroring Interfaces | 603
- RADIUS-Initiated Traffic Mirroring Process for Logged-In Subscribers | 607
- Configuring RADIUS-Initiated Subscriber Secure Policy Mirroring Overview | 616

RADIUS-Initiated Traffic Mirroring Process for Logged-In Subscribers

Figure 14 on page 607 shows the process for a RADIUS-initiated subscriber mirroring operation that is initiated after the subscriber has logged in.

Figure 14: RADIUS-Initiated Subscriber Secure Policy Model After Login



1– The subscriber logs in, requesting authentication by the RADIUS server. The RADIUS server authenticates the subscriber (no mirroring activity occurs).	6– The IAP sends the original subscriber traffic to its intended destination.
2– The LEA sends provisioning information for a subscriber whose traffic is to be mirrored over the HI-1 interface to the mediation device.	7– As subscriber-related events occur, the IAP sends the events in SNMP traps over the INI-2 interface to the mediation device.

3– The mediation device sends the provisioning information over the INI-1 interface to the RADIUS server.	8– The mediation device provides events over the HI-2 interface to the LEA.
4– The RADIUS server sends a CoA message containing the mirroring-related RADIUS attributes and VSAs to the IAP (the router).	9– The IAP encapsulates the mirrored subscriber content in a packet header and sends it to the mediation device over the INI-3 interface. The IAP uses the destination IP address that it received in the Access-Accept messaged from the RADIUS server.
5– The RADIUS CoA message initiates the mirroring operation. The IAP creates the subscriber secure policy based on the mirroring VSAs and immediately begins mirroring subscriber traffic.	10– The mediation device sends mirrored content over the HI-3 interface to the LEA.

RELATED DOCUMENTATION

- Subscriber Secure Policy Traffic Mirroring Architecture Using RADIUS | **601**
- RADIUS-Initiated Traffic Mirroring Interfaces | **603**
- RADIUS-Initiated Traffic Mirroring Process at Subscriber Login | **605**
- Configuring RADIUS-Initiated Subscriber Secure Policy Mirroring Overview | **616**

RADIUS Attributes Used for Subscriber Secure Policy

IN THIS SECTION

- Triggering Subscriber Secure Policy for Subscribers on Dynamic Authenticated VLANs | **610**

Table 42 on page 609 lists the RADIUS VSAs that are associated with subscriber secure policy. If these VSAs are present in the RADIUS Access-Accept message for a subscriber, the action specified in the LI-Action attribute takes effect.

Mirroring VSAs that the RADIUS server sends to the router are salt-encrypted. Salt encryption is a random string of data used to modify a password hash.

Table 42: RADIUS-Based Mirroring Attributes

Attribute Number	Attribute Name	Description	Value
[26-58]	LI-Action	Traffic mirroring action	Salt-encrypted integer • 0 = stop mirroring • 1 = start mirroring • 2 = no action
[26-59]	Med-Dev-Handle	Identifier that associates mirrored traffic with a specific subscriber Med-Dev-Handle includes: • Intercept-Identifier • Acct-Session-ID (optional)	Salt-encrypted string
[26-60]	Med-Ip-Address	IP address of mediation device to which mirrored traffic is forwarded	Salt-encrypted IP address
[26-61]	Med-Port-Number	UDP port in the mediation device to which mirrored traffic is forwarded	Salt-encrypted integer

NOTE: CoA-Request messages that include any of the RADIUS-based mirroring attributes (VSAs 26-58, 26-59, 26-60, or 26-61) must always include all four VSAs. If the CoA action is to stop mirroring (VSA 26-58 value is 0), then the values of the other three attributes in the CoA message must match the existing attribute values; otherwise, the action fails.

If a subscriber is already logged in, Table 43 on page 610 lists the RADIUS attributes that can be present in RADIUS CoA messages to identify the subscriber whose traffic is to have a mirroring action applied (activation or deactivation).

Table 43: RADIUS Attributes Used in CoA Messages to Identify Subscribers for Traffic Mirroring

Attribute Number	Attribute Name
[1]	User-Name
[44]	Acct-Session-ID

Triggering Subscriber Secure Policy for Subscribers on Dynamic Authenticated VLANs

BEST PRACTICE: When you have DHCPv4/DHCPv6 subscribers over VLANs, two sessions are created for each subscriber—one for the Layer 2 VLAN, and one for DHCP. In this case, we recommend that you use one trigger that matches both the DHCP and the VLAN session.

If authentication is performed on both the VLAN session and the DHCP session, we recommend that you use a separate, unique username for the VLAN and DHCP sessions to allow RADIUS to distinguish on which of the sessions to trigger subscriber secure policy traffic mirroring. Otherwise, traffic mirroring fails when the DHCP session is authenticated and activated.

RELATED DOCUMENTATION

RADIUS-Initiated Subscriber Secure Policy Overview | 600

Subscriber Secure Policy Traffic Mirroring Architecture Using RADIUS | 601

Using the Packet Header to Track Subscribers on the Mediation Device

IN THIS SECTION

- Mirror Header Format | 614
- Mirror Header Format (Junos OS Evolved with Payload Direction) | 615

When the router sends mirrored traffic to the mediation device, it encapsulates it in a packet header. Figure 15 on page 611 is the mirrored packet header and payload that the router sends to the mediation device. The packet header includes a mirror header that the mediation device can use to track subscribers and subscriber sessions. The router creates mirror header based on information that it receives from the RADIUS server.

Figure 15: Mirrored Packet Header and Payload

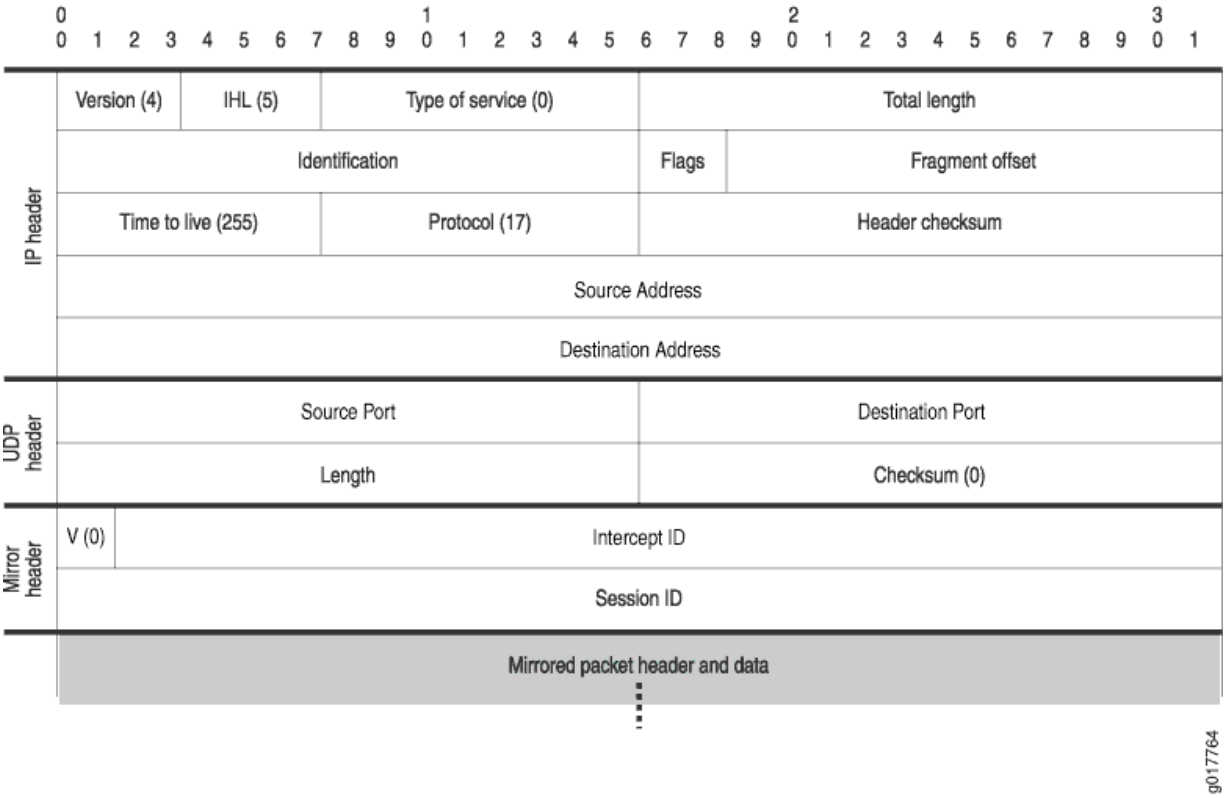


Table 44 on page 611 describes the fields in the packet header of mirrored packets.

Table 44: Mirrored Packet Header and Payload Field Descriptions For the Mediation Device

Field	Value	Length (Bits)
IP Header		
Version	4	4
IHL	5	4

Table 44: Mirrored Packet Header and Payload Field Descriptions For the Mediation Device *(Continued)*

Field	Value	Length (Bits)
Type of Service	0	8
Total Length	Dynamically computed	16
Identification	Dynamically computed	16
Flags	Dynamically computed	3
Fragment Offset	Dynamically computed	13
Time to Live	255	8
Protocol	17	8
Header Checksum	Dynamically computed	16
Source Address	IP address of the router interface that sends mirrored traffic to the mediation device	32
Destination Address	IP address of the mediation device to which mirrored traffic is forwarded (VSA 26-60)	32
UDP Header		
Source Port	UDP port number on the router from which mirrored traffic is sent to the mediation device	16

Table 44: Mirrored Packet Header and Payload Field Descriptions For the Mediation Device *(Continued)*

Field	Value	Length (Bits)
Destination Port	UDP port on the mediation device to which mirrored traffic is forwarded (VSA 26-61)	16
Length	Dynamically computed	16
Checksum	0	16
Mirror Header		
V (mirror header value)	Used by the router to specify how the values of the Session ID and Intercept ID are determined. The value received from RADIUS can be 0 or 1. However, the value is always 0 in the packet header sent to the mediation device.	2
Intercept ID	Used along with the Session ID by the mediation device to track a subscriber across multiple login and logout events. The value is assigned to a subscriber whose traffic is being intercepted. The Intercept ID is constant; it does not change as a subscriber logs in and logs out of sessions.	30
Session ID	Used by the mediation device to identify the session of the mirrored subscriber. The value is assigned to a subscriber session by the Junos OS. The Session ID changes with each new session for a subscriber.	32

Mirror Header Format

The values of the Intercept ID and the Session ID are determined by the value that the router receives in VSA 26-59. VSA 26-59 is declared as a hexadecimal string that can be either 4 bytes or 8 bytes long. The mirror header value specifies whether a 4-byte value or an 8-byte value is used to form the Intercept ID and the Session ID.

The VSA 26-59 is a hexadecimal format. For example, a value of 40000010 becomes 0x40000010 in VSA 26-59 format. V = 0 is denoted as V = 0x0 in hexadecimal format.

4-Byte Format

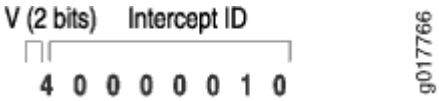
The 4-byte format allows you to manually specify the Intercept ID. The Session ID value is automatically created based on the least significant 32 bits of the Acct-Session-ID (RADIUS attribute 44).

To use the 4-byte format of VSA 26-59, configure the first two most significant bits of the VSA to the hexadecimal value of 01, where the 1st bit is 0 and the 2nd bit is 1. The 4-byte format indicates a single word in the VSA. The remaining 30 bits of the word form the Intercept ID value.

For example, a value of 40000010 for VSA 26-59 configures the following fields in the mirror header, as shown in Figure 16 on page 614:

- V = 1
- Intercept ID = 0x10

Figure 16: 4-Byte Format of VSA 26-59



8-Byte Format

The 8-byte format of VSA 26-59 enables you to manually specify the both the Session-ID value and the Intercept ID value.

To use the 8-byte format, you configure the first two most significant bits of the first word of the VSA to a value of 0, which indicates two words in the VSA. The remaining 30 bits of the first word form the Intercept ID value, and the second word is the Session-ID field. You cannot change the order of these two words.

For example, a value of 0000030000000090 in VSA 26-59 configures the following fields in the mirror header, as shown in Figure 17 on page 615:

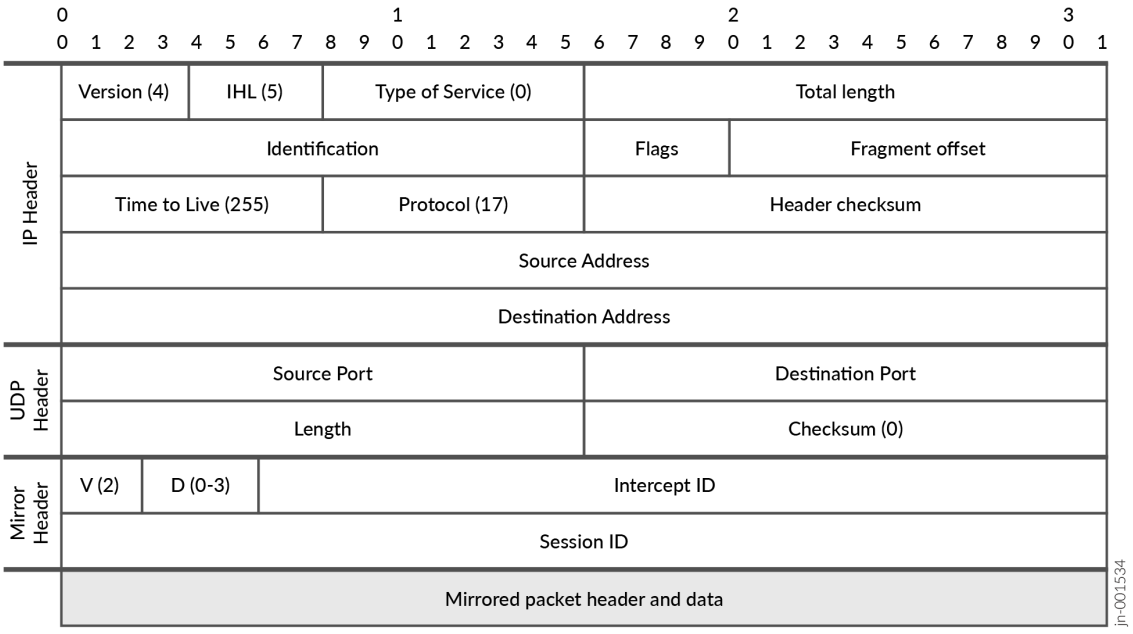
- V = 0
- Intercept ID = 0x300
- Session ID = 0x90

Figure 17: 8-Byte Format of VSA 26-59

Mirror Header Format (Junos OS Evolved with Payload Direction)

In Junos OS Evolved devices, you can enable Direction(D) field in the mirror header by using the `set system services radius-flow-tap extended-mirror-header` command. The Direction(D) field helps the mediation device to quickly identify the traffic direction and apply the appropriate rules defined by the law enforcement agency.

Figure 18: Mirrored Packet Header and Payload with Direction Field



Configure the first two most significant bits of the VSA to a value of 2 to indicate that the mirror header contains the Direction field, See Mirror Header in Figure 18 on page 615.

The Direction(D) field is 3-bits long. The possible values are as follows:

- 0: Reserved for keepalive mechanism
- 1: The direction of the intercepted data or event is not known to the Point of Interconnect (POI)
- 2: Intercepted data or event was sent to target (downstream)
- 3: Intercepted data or event was sent to target (upstream)

RELATED DOCUMENTATION

RADIUS-Initiated Subscriber Secure Policy Overview | 600

Subscriber Secure Policy Traffic Mirroring Architecture Using RADIUS | 601

Configuring RADIUS-Initiated Subscriber Secure Policy Mirroring Overview

Before you configure subscriber secure policy traffic mirroring, note the following:

- Subscriber secure policy mirroring runs on the radius-flow-tap service infrastructure. To configure the subscriber secure policy service, you must have the same privileges that are required to configure the radius-flow-tap service.
- The subscriber secure policy feature requires some system resources while mirroring, encrypting, and sending traffic to the mediation device. For example, you might elect to use a 10-Gigabit Ethernet interface for the tunnel to the mediation device if you expect the amount of traffic you plan to mirror to approach 1 Gbps of actual user data.

To configure the subscriber secure policy service:

1. Configure radius-flow-tap service support for secure subscriber policy. This support includes optional forwarding-class information that the subscriber secure policy service uses to send mirrored traffic to the content destination device.

   See "Configuring Support for Subscriber Secure Policy Mirroring" on page 619.
2. Configure an access profile that specifies the RADIUS-related support for subscriber secure policy on the router, including a list of one or more RADIUS authentication servers. The router uses the list of specified servers for both authentication and dynamic request operations. You must also configure

the RADIUS dynamic request feature, which provides the CoA message support used in-session traffic mirroring.

See "Configuring RADIUS Server Support for Subscriber Secure Policy Mirroring" on page 622.

3. Ensure that the following support is also configured:
   - The RADIUS record of the mirrored subscriber must include the RADIUS attributes and VSAs required for subscriber secure policy mirroring. See "RADIUS Attributes Used for Subscriber Secure Policy" on page 608 for descriptions of the supported attributes used in RADIUS Accept-Accept and CoA messages.
   - The mediation device must be configured to accept the mirrored content.
4. (Optional) Enable the mirroring of IPv4 multicast traffic on the router.

   See "Enabling Subscriber Secure Policy Mirroring for IPv4 Multicast Traffic" on page 673.
5. (Optional) Configure SNMPv3 trap support to report mirroring-related events to the mediation device.

   See "Configuring SNMPv3 Traps for Subscriber Secure Policy Mirroring" on page 678.

You can terminate an active subscriber mirroring session at any time.

See "Terminating RADIUS-Initiated Subscriber Traffic Mirroring " on page 623.

RELATED DOCUMENTATION

RADIUS Attributes Used for Subscriber Secure Policy | 608

Guidelines for Configuring Subscriber Secure Policy Mirroring | 617

Intercept-Related Events Transmitted to the Mediation Device | 675

Terminating RADIUS-Initiated Subscriber Traffic Mirroring | 623

Guidelines for Configuring Subscriber Secure Policy Mirroring

The subscriber secure policy service uses the radius-flow-tap service infrastructure. Consider the following guidelines when you configure subscriber secure policy mirroring:

When configuring subscriber secure policy mirroring, consider the following guidelines regarding the relationship between the radius-flow-tap service and the FlowTapLite service on MX Series tunnel interfaces (FlowTapLite):

- Starting in Junos OS Release 17.3R1, the radius-flow-tap service can run concurrently on the same router with the FlowTapLite service. The FlowTapLite service is a version of the flow-tap service ([edit

`services flow-tap`]) that is configured only on tunnel interfaces on MX Series routers and is not used for subscriber secure policy mirroring.

In earlier releases, the radius-flow-tap and FlowTapLite services cannot run concurrently on an MX Series router, preventing you from running FlowTapLite monitoring and subscriber secure policy mirroring at the same time.

- You can configure one instance of the radius-flow-tap service on the router. Subscriber secure policy RADIUS-initiated mirroring and Dynamic Tasking Control Protocol (DTCP)-initiated mirroring both use the radius-flow-tap service.
- If you delete the radius-flow-tap service, new subscribers are not monitored. Existing subscribers that already have subscriber secure policy attached are not affected when you delete the service configuration.
- You can retain DTCP-initiated mirroring but prevent RADIUS-initiated mirroring from being enabled by including the `[edit system services dtcp-only]` statement, if you do so before any RADIUS-initiated mirroring is attached to a subscriber. Subsequently, RADIUS requests to initiate mirroring are rejected; only DTCP-initiated mirroring and FlowTapLite are allowed. Existing RADIUS-initiated mirroring services are not affected.
- Starting in Junos OS Release 16.1R1, you must configure the target parameters for mediation devices so that the SNMPv3 traps are sent with privacy (encrypted). Targets without privacy configured cannot receive the notifications. In earlier releases, you can configure target parameters without privacy, allowing unencrypted notifications to be sent to the mediation devices. You must also explicitly configure a list of trap targets with the `[edit services radius-flow-tap snmp notify-targets]` statement.

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
17.3R1	Starting in Junos OS Release 17.3R1, the radius-flow-tap service can run concurrently on the same router with the FlowTapLite service.
16.1R1	Starting in Junos OS Release 16.1R1, you must configure the target parameters for mediation devices so that the SNMPv3 traps are sent with privacy (encrypted).

RELATED DOCUMENTATION

Subscriber Secure Policy Overview | 595

Configuring RADIUS-Initiated Subscriber Secure Policy Mirroring Overview | 616
Configuring DTCP-Initiated Subscriber Secure Policy Mirroring Overview | 636
Configuring Support for Subscriber Secure Policy Mirroring | 619
Disabling RADIUS-Initiated Subscriber Secure Policy Mirroring | 644

Configuring Support for Subscriber Secure Policy Mirroring

Subscriber secure policy runs on the radius-flow-tap service. This topic describes the steps to configure radius-flow-tap support for RADIUS-initiated and DTCP-initiated subscriber secure policy mirroring.

To configure the radius-flow-tap service to support subscriber secure policy mirroring:

1. Configure the flow-tap service used for subscriber secure policy mirroring.

```
[edit services]
user@host# edit radius-flow-tap
```

2. Specify how the mirrored packets are forwarded to the mediation device.

> **NOTE**: The actions in this step vary based on whether you're using extensible subscriber services manager (ESSM). When using ESSM you define a virtual tunnel (vt) interface that is placed into a routing instance. ESSM determines the routing instance for the flow tap based on this vt interface. When not using ESSM the routing instance used for the tap is explicitly configured under the `services radius-flow-tap` hierarchy.

- If ESSM is used to managed the tapped subscriber interface:

  Define a vt interface. You only perform this action when the tapped interfaces are managed by extensible subscriber services manager (ESSM).

```
[edit services radius-flow-tap]
user@host# set interfaces vt-1/1/0.0
```

  If a currently used tunnel interface is deleted from the pool of interfaces, the active mirroring sessions are redistributed from the deleted interface to other tunnel interfaces in the pool. Also, when a new tunnel interface is added into the pool, the service adds the new interface to the list of interfaces available for new mirroring sessions or for existing sessions transferred from a failed interface.

- If EESM is not used to manage the tapped subscriber interface:

  Specify the logical system and routing instance for the `radius-flow-tap` service. When not using EESM a vt interface is *not* required.

  ```
  [edit services radius-flow-tap]
  user@host# set logical-system LS1 routing-instance RI1
  ```

  You can specify a logical system and routing instance, or a routing instance without a logical system. If you do not specify a logical system, the router uses logical system `default`. If you do not specify either a logical system or routing instance, the router uses logical system `default` and routing instance `default`.

  BEST PRACTICE: Configure a routing instance to prevent a spoofed mediation device address from diverting traffic away from the device. When the mirrored customer flows are in the same routing instance as the mediation device, a malicious user might hijack the mediation device's route advertisement. By advertising a next hop to the hijacker's network instead of to the device, the mirrored flows are captured and never reach the mediation device.

  If you configure the mirrored traffic to be forwarded to the mediation device by means of a routing instance, then the traffic is separated from the Internet. An external user is then unable to divert the mirrored traffic to the user' s network.

  NOTE: The `interfaces` statement applies only to ESSM-created interfaces and is ignored for flow-based interfaces. Similarly, the LS:RI configuration applies only to flow-based interfaces.

3. Specify the source IP address that the radius-flow-tap service uses for mirroring. This address is used in the IP header prepended to mirrored packets that are sent to the content destination device.

   ```
   [edit services radius-flow-tap]
   user@host# set source-ipv4-address ipv4-address
   ```

4. (Optional) Specify the forwarding class that is applied to the mirrored packets sent to the mediation device.

If you do not specify a forwarding class, mirrored packets inherit the forwarding class from the original packet (which is the forwarding class set by default classification that CoS applies to the packet on the ingress interface).

```
[edit services radius-flow-tap]
user@host# set forwarding-class class-name
```

5. (Optional) Specify the subscriber secure policy that determines what traffic, if any, is not sent to the mediation device.

```
[edit services radius-flow-tap]
user@host# set policy policy-name
```

NOTE: You can add or change a subscriber secure policy any time, but a changed policy does not apply to a currently enabled policy. To change a policy:

- Send a DTCP DELETE message to remove the current policy.
- Modify the configuration with the new version of the policy.
- Send a DTCP ADD message to add the policy.
- Send a DTCP ENABLE message to enable the policy.

6. (Optional) Specify the IP address for one or more target mediation devices to receive SNMPv3 trap notifications. Each target address must be configured separately.

```
[edit services radius-flow-tap]
user@host# set snmp notify-targets ip-address
```

NOTE: You must also configure SNMP so that only encrypted notifications are sent to target devices. Targets without privacy configured cannot receive the notifications. For information about the SNMP configuration for subscriber secure policy, see "Configuring SNMPv3 Traps for Subscriber Secure Policy Mirroring" on page 678.

RELATED DOCUMENTATION

Extensible Subscriber Services Manager Overview

Subscriber Secure Policy Overview | 595
Configuring RADIUS-Initiated Subscriber Secure Policy Mirroring Overview | 616
Configuring DTCP-Initiated Subscriber Secure Policy Mirroring Overview | 636
Guidelines for Configuring Subscriber Secure Policy Mirroring | 617

Configuring RADIUS Server Support for Subscriber Secure Policy Mirroring

This topic describes how to configure support for the RADIUS server that initiates subscriber-based traffic mirroring. You create an access profile to specify the RADIUS server support.

To configure the router's interaction with the RADIUS server in support of subscriber secure policy mirroring:

1. Create the access profile and assign a name.

```
[edit access]
user@host# edit profile profile-name
```

2. Specify RADIUS as the authentication method.

```
[edit access profile profile-name]
user@host# set authentication-order radius
```

3. Specify the IP address of the RADIUS server that performs authentication. This server also performs dynamic request (CoA) functions.

```
[edit access profile profile-name]
user@host# set radius authentication-server ip-address
```

4. Specify the secret to use when communicating with the RADIUS server.

```
[edit access profile profile-name]
user@host# set radius-server server-address secret password
```

5. Specify other optional RADIUS configuration settings as needed, such as accounting support.

RELATED DOCUMENTATION

Subscriber Secure Policy Overview | 595

Configuring RADIUS-Initiated Subscriber Secure Policy Mirroring Overview | 616

RADIUS Attributes Used for Subscriber Secure Policy | 608

Terminating RADIUS-Initiated Subscriber Traffic Mirroring

You can terminate RADIUS-initiated traffic mirroring sessions by the following action:

- RADIUS CoA message receipt—Terminated upon receipt of a CoA message with the VSA 26-58 (LI-Action) value of 0. The RADIUS administrator configures the LI-Action of 0 in the mirrored subscriber's RADIUS record.

RELATED DOCUMENTATION

RADIUS-Initiated Subscriber Secure Policy Overview | 600

Configuring RADIUS-Initiated Subscriber Secure Policy Mirroring Overview | 616

CHAPTER 36

Configuring DTCP-Initiated Subscriber Secure Policy Traffic Mirroring

IN THIS CHAPTER

- DTCP-Initiated Subscriber Secure Policy Overview | 624
- Subscriber Secure Policy Traffic Mirroring Architecture Using DTCP | 625
- DTCP-Initiated Traffic Mirroring Interfaces | 627
- DTCP-Initiated Traffic Mirroring Process | 628
- DTCP Messages Used for Subscriber Secure Policy | 630
- Packet Header for Mirrored Traffic Sent to Mediation Device | 631
- Configuring DTCP-Initiated Subscriber Secure Policy Mirroring Overview | 636
- Guidelines for Configuring Subscriber Secure Policy Mirroring | 637
- Configuring Support for Subscriber Secure Policy Mirroring | 638
- Configuring the Mediation Device as a User on the Router | 641
- Configuring a DTCP-over-SSH Connection to the Mediation Device | 643
- Configuring the Mediation Device to Provision Traffic Mirroring | 644
- Disabling RADIUS-Initiated Subscriber Secure Policy Mirroring | 644
- Example: Configuring Traffic That Is Mirrored Using DTCP-Initiated Subscriber Secure Policy | 645
- Terminating DTCP-Initiated Subscriber Traffic Mirroring Sessions | 648

DTCP-Initiated Subscriber Secure Policy Overview

Dynamic Tasking Control Protocol (DTCP)-initiated mirroring creates secure policies to mirror traffic for the subscriber based on DTCP messages. The attributes in a DTCP ADD message sent from the mediation device trigger the router to start mirroring traffic and specify the interface on which the mirroring takes place. The mirroring operations can be initiated by DTCP messages as follows:

- Subscriber login—Mirroring starts on the specified interface when the subscriber logs in. The DTCP ADD message must be sent to the router before the subscriber logs in.

- In-session—Mirroring starts for all subscribers that match the trigger supplied in the DTCP ADD message when the router receives a DTCP ADD message.

RELATED DOCUMENTATION

Subscriber Secure Policy Traffic Mirroring Architecture Using DTCP | 625

Configuring DTCP-Initiated Subscriber Secure Policy Mirroring Overview | 636

Subscriber Secure Policy Traffic Mirroring Architecture Using DTCP

Figure 19 on page 625 shows the architecture of the DTCP-initiated subscriber secure policy mirroring environment.

Figure 19: DTCP-Initiated Subscriber Secure Policy Architecture

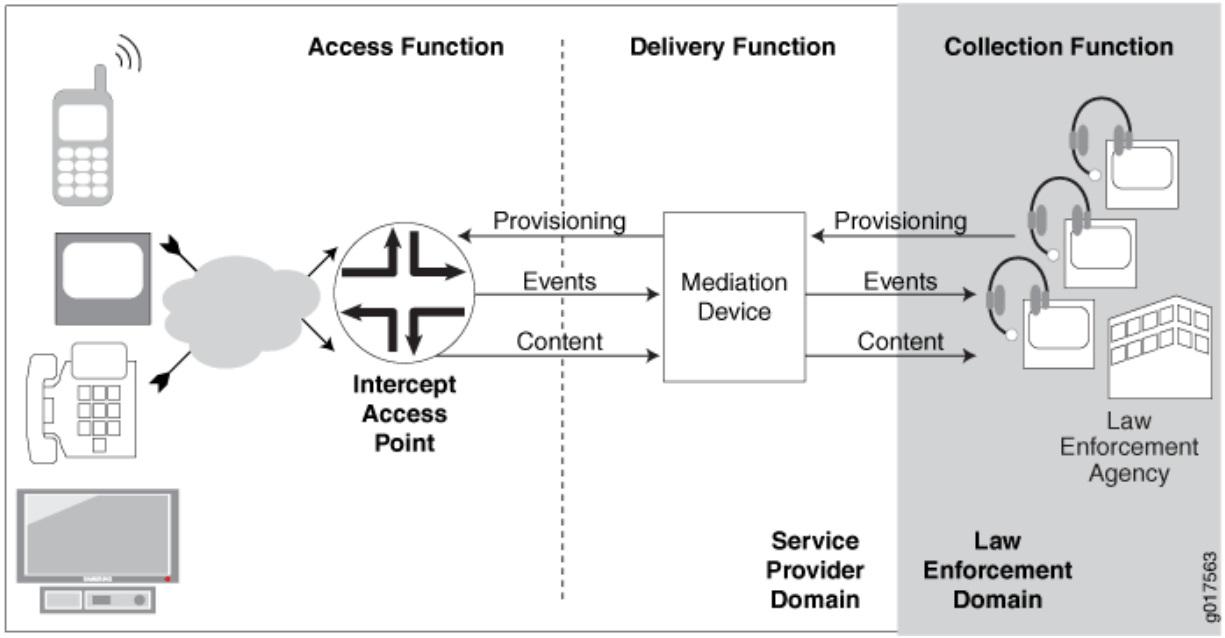


Table 45 on page 626 describes the functions and components of a DTCP-initiated subscriber secure policy traffic mirroring environment.

Table 45: DTCP-Initiated Subscriber Secure Policy Functions and Components

Function or Component	Description
Collection function	The collection function is responsible for collecting intercepted content and identifying information from the delivery function. The collection function is the responsibility of the law-enforcement agency (LEA).
Delivery function	The delivery function delivers information that it receives from the access function to the collection function. The delivery function is performed by the mediation device.
Access function	The access function has access to the intercept target's traffic content and intercept-related events. It is responsible for collecting this information and sending it to the delivery function. The access function is performed by intercept access points (IAPs).
Events	Intercept-related events, such as login or logout events or mirroring session activation or deactivation. The router sends the events to the mediation device in SNMP traps.
LEA	Law enforcement agency. The LEA provides intercept targets to the service provider who provisions the mediation device.
Mediation device	The mediation device receives provisioning information from the LEA, and it uses the information to send provisioning information to the IAP (the router). The mediation device also receives intercept-related events and intercepted content from the router, and delivers the events and content to the LEA.
IAP	Intercept access point. In a subscriber access network the Juniper Networks router is the IAP. Using subscriber secure policies, the IAP intercepts traffic to and from the subscriber whose traffic is being mirrored. It encapsulates the intercepted content in a packet header and delivers it to the mediation device, while also sending the traffic to the intended destination. The IAP also sends intercept-related events to the mediation device using SNMP traps.

RELATED DOCUMENTATION

DTCP-Initiated Subscriber Secure Policy Overview | 624

DTCP-Initiated Traffic Mirroring Interfaces | 627

DTCP-Initiated Traffic Mirroring Process | 628

DTCP-Initiated Traffic Mirroring Interfaces

Figure 20 on page 627 shows the interfaces involved in DTCP-initiated secure subscriber policy traffic mirroring.

Figure 20: DTCP-Initiated Traffic Mirroring Interfaces

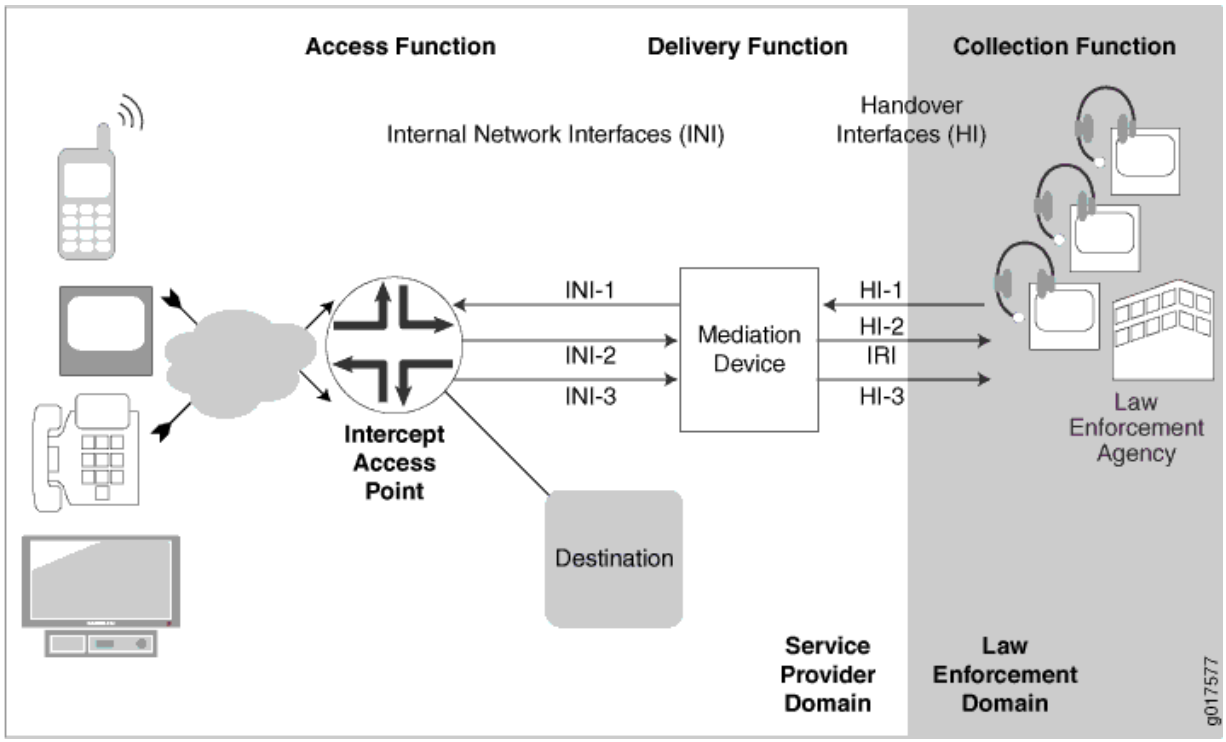


Table 46 on page 628 describes the interfaces involved in DTCP-initiated secure subscriber policy traffic mirroring.

Table 46: DTCP-Initiated Traffic Mirroring Interfaces

Interface	Description
HI-1	Handover Interface 1—Administrative interface between the LEA and the service provider mediation device. The LEA sends provisioning information to the mediation device on this interface.
HI-2	Handover Interface 2—Intercept-related information interface between the LEA and the mediation device that is used to deliver intercept-related events to the LEA. These events can be subscriber session events such as login, logout, and authentication.
HI-3	Handover Interface 3—Intercepted content interface between the mediation device and LEA that is used to deliver intercepted content to the LEA.
INI-1	Internal network Interface 1—Interface used to send DTCP messages containing intercept provisioning information from the mediation device to the router.
INI-2	Internal network interface 2—Interface used to send intercept-related events from the router to the mediation device. This information is sent in SNMP traps.
INI-3	Internal network interface 3—Interface used to send intercepted content from the router to the mediation device.

RELATED DOCUMENTATION

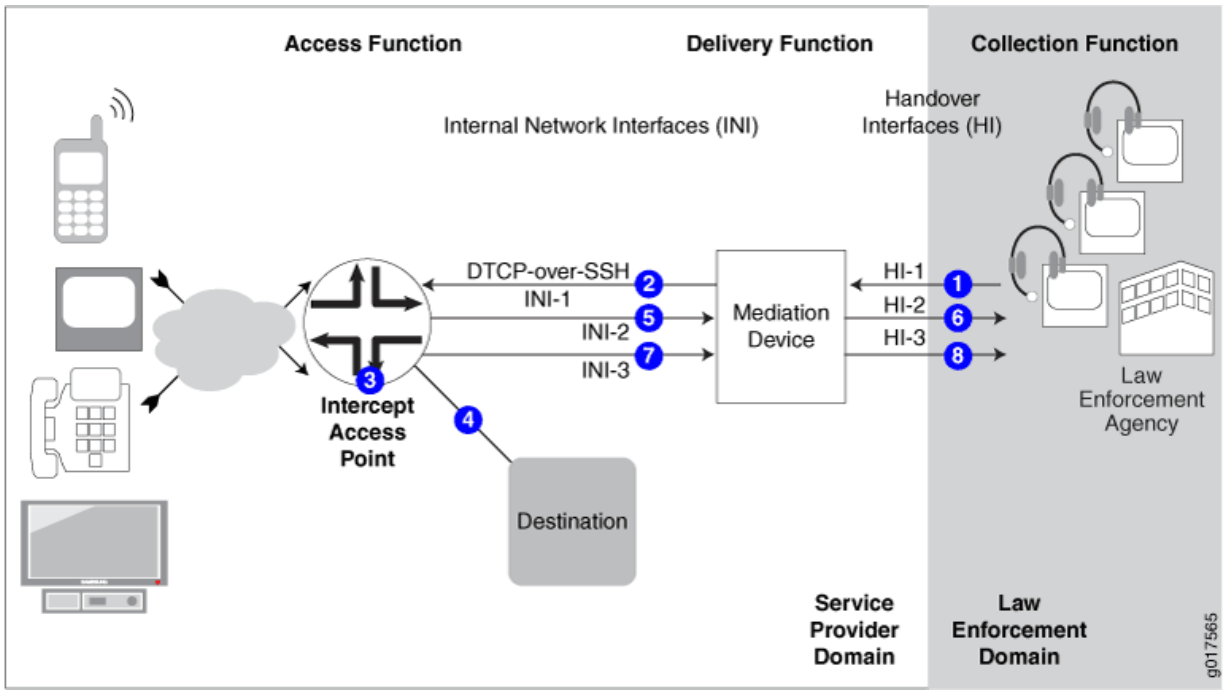
Subscriber Secure Policy Traffic Mirroring Architecture Using DTCP | 625

DTCP-Initiated Traffic Mirroring Process | 628

DTCP-Initiated Traffic Mirroring Process

Figure 21 on page 629 shows the process for a DTCP-initiated subscriber mirroring operation.

Figure 21: DTCP-Initiated Subscriber Secure Policy Model



1– The LEA sends provisioning information for a subscriber whose traffic is to be mirrored over the HI-1 interface to the mediation device.	5– As intercept-related events occur, the IAP sends the events in SNMP traps over the INI-2 interface to the mediation device.
2– The mediation device sends a DTCP ADD message that contains provisioning information over the INI-1 interface to the IAP (the router).	6– The mediation device provides the intercept-related events over the HI-2 interface to the LEA.
3– The IAP creates a subscriber secure policy based on information in the DTCP ADD message. If the IAP receives the DTCP ADD before the subscriber logs in, mirroring begins when the subscriber logs in. If the router receives the DTCP ADD after the subscriber logs in, mirroring begins when the ADD message is received.	7– The IAP sends the mirrored content to the mediation device over the INI-3 interface.
4– The IAP sends the original subscriber traffic to its intended destination.	8– The mediation device sends mirrored content over the HI-3 interface to the LEA.

RELATED DOCUMENTATION

Subscriber Secure Policy Traffic Mirroring Architecture Using DTCP | 625

DTCP-Initiated Traffic Mirroring Interfaces | 627

DTCP Messages Used for Subscriber Secure Policy | 630

DTCP Messages Used for Subscriber Secure Policy

You can use DTCP to provision traffic mirroring on the router by sending DTCP messages from the mediation device to the router.

There are four types of DTCP messages supported for radius-flow-tap services:

- ADD—Triggers mirroring of subscriber secure policy sessions. You include attributes that trigger the router to begin mirroring a subscriber session. In addition to one or more attributes that trigger the router to begin traffic mirroring, you can also include attributes that identify where to send the mirrored session data and how to uniquely identify traffic when simultaneous intercepts are active. The ADD message also provides instructions to populate fields in the encapsulation header for packets sent to the mediation device.
- DELETE—Removes a subscriber mirroring trigger or can be used to remove all mirroring.
- ENABLE—Triggers a drop policy on the router if one does not already exist from a prior DTCP ADD or DTCP ENABLE message.
- LIST—Requests information about sessions that are currently being mirrored. This information is returned in a LIST response.

NOTE: Consult the documentation for your mediation device to learn how to configure DTCP messages on the device.

RELATED DOCUMENTATION

DTCP-Initiated Traffic Mirroring Process | 628

ADD (DTCP)

DELETE (DTCP)

ENABLE (DTCP)

LIST (DTCP)

Packet Header for Mirrored Traffic Sent to Mediation Device

IN THIS SECTION

- Format of the Mirror Header Values Used to Track Subscribers and Subscriber Sessions | **634**
- Manually Setting the Session-ID and Intercept ID in Packet Headers | **635**

When the router sends mirrored traffic to the mediation device, it encapsulates the mirrored payload in a packet header before it sends the mirrored traffic to the mediation device.

The packet header includes the Session ID that Junos assigns to the subscriber session. The mediation device can use the ID to identify the session of the mirrored subscriber. The mediation device can use the Session ID along with the Intercept ID to track a subscriber across multiple login and logout events. The Intercept ID is constant, but the Session ID changes with each new session for a subscriber.

Figure 22 on page 632 is the mirrored packet header that the router sends to the mediation device.

Figure 22: Mirrored Packet Header and Payload

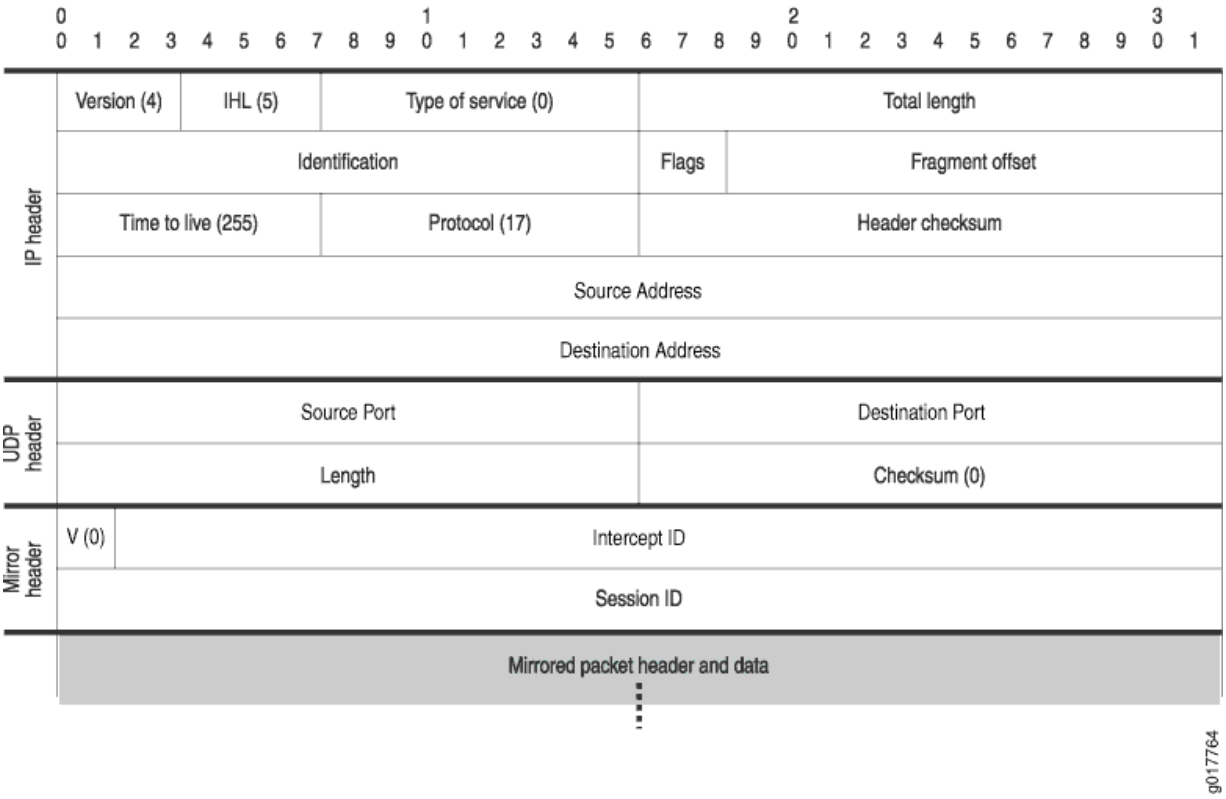


Table 47 on page 632 describes the fields in the packet header of mirrored packets.

Table 47: Packet Header Field Descriptions

Field	Value	Length (Bits)
IP Header		
Version	4	4
IHL	5	4
Type of Service	0	8
Total Length	Dynamically computed	16

Table 47: Packet Header Field Descriptions *(Continued)*

Field	Value	Length (Bits)
Identification	Dynamically computed	16
Flags	Dynamically computed	3
Fragment Offset	Dynamically computed	13
Time to Live	255	8
Protocol	17	8
Header Checksum	Dynamically computed	16
Source Address	IP address of the router interface that sends mirrored traffic to the mediation device	32
Destination Address	IP address of the mediation device to which mirrored traffic is forwarded. This value is taken from the X-JTap-Cdest-Dest-Address attribute that is sent to the router in the DTCP ADD command.	32
UDP Header		
Source Port	UDP port number on the router from which mirrored traffic is sent to the mediation device	16

Table 47: Packet Header Field Descriptions *(Continued)*

Field	Value	Length (Bits)
Destination Port	UDP port on the mediation device to which mirrored traffic is forwarded. This value is taken from the X-JTap-Cdest-Dest-Port attribute that is sent to the router in the DTCP ADD command.	16
Length	Dynamically computed	16
Checksum	0	16
Mirror Header		
V (mirror header value)	0	2
Intercept ID	Value of the X-MD-Intercept-Id that is sent to the router in the DTCP ADD command.	30
Session ID	Subscriber session ID assigned by the router. See "Format of the Mirror Header Values Used to Track Subscribers and Subscriber Sessions" on page 634	32

Format of the Mirror Header Values Used to Track Subscribers and Subscriber Sessions

The packet header includes mirror header attributes that the mediation device can use to track subscribers and subscriber sessions. There are three mirror header attributes in the packet header:

- V (mirror header value)— For DTCP, this value is always set to 0 in the packet header sent to the mediation device.

- Session ID— Used by the mediation device to identify the session of the mirrored subscriber. The value is assigned to a subscriber session by the Junos OS. The Session ID changes with each new session for a subscriber.
- Intercept ID— Used along with the Session ID by the mediation device to track a subscriber across multiple login and logout events. The value is assigned to a subscriber whose traffic is being intercepted. The Intercept ID is constant; it does not change as a subscriber logs in and logs out of sessions.

Manually Setting the Session-ID and Intercept ID in Packet Headers

You can use the DTCP ADD command to manually specify the Session ID value (`X-Act-Sess-Id`) and the Intercept ID value (X-MD-Intercept-Id) placed in the headers sent to the mediation device. You configure the values in an 8-byte format. To do so:

- Configure the first two most significant bits to a value of 0, which indicates two words.
- Configure the remaining 30 bits of the first word to form the Intercept ID field.
- Configure the second word to form the Session-ID field.

You cannot change the order of these two words.

Figure 23 on page 635 shows an example of the mirror header:

Figure 23: Mirror Header Format

For example, a value of 0000030000000090 configures the following fields in the mirror header: :

- V = 0
- Intercept-ID = 0x300
- Session-ID = 0x90

RELATED DOCUMENTATION

DTCP-Initiated Subscriber Secure Policy Overview | 624

ADD (DTCP)

Example: Using DTCP Messages to Trigger, Verify, and Remove Traffic Mirroring for Subscribers | 665

Configuring DTCP-Initiated Subscriber Secure Policy Mirroring Overview

Before you configure subscriber secure policy traffic mirroring, note the following:

- Subscriber secure policy mirroring runs on the radius-flow-tap service infrastructure. To configure the subscriber secure policy service, you need the same privileges that are required to configure the radius-flow-tap service.
- The subscriber secure policy feature requires some system resources while mirroring, encrypting, and sending traffic to the mediation device. For example, you might elect to use a 10-Gigabit Ethernet interface for the tunnel and mediation device if you expect the amount of traffic you plan to mirror to approach 1 Gbps of actual user data.

To configure DTCP-initiated subscriber secure policy service:

1. Configure the radius-flow-tap service support for secure subscriber policy. This support includes configuring the tunnels and optional forwarding-class information that the subscriber secure policy service uses to send mirrored traffic to the content destination device.

   See "Configuring Support for Subscriber Secure Policy Mirroring" on page 619.
2. Configure the mediation device as a user on the router. This user account allows the router to receive DTCP messages from the mediation device.

   See "Configuring the Mediation Device as a User on the Router" on page 641.
3. Configure the mediation device to provision traffic mirroring on the router.

   See "Configuring the Mediation Device to Provision Traffic Mirroring" on page 644.
4. Configure a DTCP-over-SSH connection to the mediation device.

   See "Configuring a DTCP-over-SSH Connection to the Mediation Device" on page 643.
5. (Optional) Enable mirroring of IPv4 multicast traffic on the router.

   See "Enabling Subscriber Secure Policy Mirroring for IPv4 Multicast Traffic" on page 673
6. Configure SNMPv3 trap support to report mirroring information to an external device.

   See "Configuring SNMPv3 Traps for Subscriber Secure Policy Mirroring" on page 678.

You can terminate an active subscriber mirroring session at any time.

See "Terminating DTCP-Initiated Subscriber Traffic Mirroring Sessions" on page 648.

RELATED DOCUMENTATION

DTCP-Initiated Subscriber Secure Policy Overview | 624

Intercept-Related Events Transmitted to the Mediation Device | 675

Guidelines for Configuring Subscriber Secure Policy Mirroring

The subscriber secure policy service uses the radius-flow-tap service infrastructure. Consider the following guidelines when you configure subscriber secure policy mirroring:

When configuring subscriber secure policy mirroring, consider the following guidelines regarding the relationship between the radius-flow-tap service and the FlowTapLite service on MX Series tunnel interfaces (FlowTapLite):

- Starting in Junos OS Release 17.3R1, the radius-flow-tap service can run concurrently on the same router with the FlowTapLite service. The FlowTapLite service is a version of the flow-tap service (`[edit services flow-tap]`) that is configured only on tunnel interfaces on MX Series routers and is not used for subscriber secure policy mirroring.

  In earlier releases, the radius-flow-tap and FlowTapLite services cannot run concurrently on an MX Series router, preventing you from running FlowTapLite monitoring and subscriber secure policy mirroring at the same time.

- You can configure one instance of the radius-flow-tap service on the router. Subscriber secure policy RADIUS-initiated mirroring and Dynamic Tasking Control Protocol (DTCP)-initiated mirroring both use the radius-flow-tap service.

- If you delete the radius-flow-tap service, new subscribers are not monitored. Existing subscribers that already have subscriber secure policy attached are not affected when you delete the service configuration.

- You can retain DTCP-initiated mirroring but prevent RADIUS-initiated mirroring from being enabled by including the `[edit system services dtcp-only]` statement, if you do so before any RADIUS-initiated mirroring is attached to a subscriber. Subsequently, RADIUS requests to initiate mirroring are rejected; only DTCP-initiated mirroring and FlowTapLite are allowed. Existing RADIUS-initiated mirroring services are not affected.

- Starting in Junos OS Release 16.1R1, you must configure the target parameters for mediation devices so that the SNMPv3 traps are sent with privacy (encrypted). Targets without privacy configured cannot receive the notifications. In earlier releases, you can configure target parameters without privacy, allowing unencrypted notifications to be sent to the mediation devices. You must also explicitly configure a list of trap targets with the `[edit services radius-flow-tap snmp notify-targets]` statement.

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
17.3R1	Starting in Junos OS Release 17.3R1, the radius-flow-tap service can run concurrently on the same router with the FlowTapLite service.
16.1R1	Starting in Junos OS Release 16.1R1, you must configure the target parameters for mediation devices so that the SNMPv3 traps are sent with privacy (encrypted).

RELATED DOCUMENTATION

- Subscriber Secure Policy Overview | 595
- Configuring RADIUS-Initiated Subscriber Secure Policy Mirroring Overview | 616
- Configuring DTCP-Initiated Subscriber Secure Policy Mirroring Overview | 636
- Configuring Support for Subscriber Secure Policy Mirroring | 619
- Disabling RADIUS-Initiated Subscriber Secure Policy Mirroring | 644

Configuring Support for Subscriber Secure Policy Mirroring

Subscriber secure policy runs on the radius-flow-tap service. This topic describes the steps to configure radius-flow-tap support for RADIUS-initiated and DTCP-initiated subscriber secure policy mirroring.

To configure the radius-flow-tap service to support subscriber secure policy mirroring:

1. Configure the flow-tap service used for subscriber secure policy mirroring.

```
[edit services]
user@host# edit radius-flow-tap
```

2. Specify how the mirrored packets are forwarded to the mediation device.

NOTE: The actions in this step vary based on whether you're using extensible subscriber services manager (ESSM). When using ESSM you define a virtual tunnel (vt) interface that is placed into a routing instance. ESSM determines the routing instance

for the flow tap based on this vt interface. When not using ESSM the routing instance used for the tap is explicitly configured under the `services radius-flow-tap` hierarchy.

- If ESSM is used to managed the tapped subscriber interface:

  Define a vt interface. You only perform this action when the tapped interfaces are managed by extensible subscriber services manager (ESSM).

```
[edit services radius-flow-tap]
user@host# set interfaces vt-1/1/0.0
```

  If a currently used tunnel interface is deleted from the pool of interfaces, the active mirroring sessions are redistributed from the deleted interface to other tunnel interfaces in the pool. Also, when a new tunnel interface is added into the pool, the service adds the new interface to the list of interfaces available for new mirroring sessions or for existing sessions transferred from a failed interface.

- If EESM is not used to manage the tapped subscriber interface:

  Specify the logical system and routing instance for the `radius-flow-tap` service. When not using EESM a vt interface is *not* required.

```
[edit services radius-flow-tap]
user@host# set logical-system LS1 routing-instance RI1
```

  You can specify a logical system and routing instance, or a routing instance without a logical system. If you do not specify a logical system, the router uses logical system `default`. If you do not specify either a logical system or routing instance, the router uses logical system `default` and routing instance `default`.

BEST PRACTICE: Configure a routing instance to prevent a spoofed mediation device address from diverting traffic away from the device. When the mirrored customer flows are in the same routing instance as the mediation device, a malicious user might hijack the mediation device's route advertisement. By advertising a next hop to the hijacker's network instead of to the device, the mirrored flows are captured and never reach the mediation device.

If you configure the mirrored traffic to be forwarded to the mediation device by means of a routing instance, then the traffic is separated from the Internet. An external user is then unable to divert the mirrored traffic to the user' s network.

NOTE: The `interfaces` statement applies only to ESSM-created interfaces and is ignored for flow-based interfaces. Similarly, the LS:RI configuration applies only to flow-based interfaces.

3. Specify the source IP address that the radius-flow-tap service uses for mirroring. This address is used in the IP header prepended to mirrored packets that are sent to the content destination device.

```
[edit services radius-flow-tap]
user@host# set source-ipv4-address ipv4-address
```

4. (Optional) Specify the forwarding class that is applied to the mirrored packets sent to the mediation device.

   If you do not specify a forwarding class, mirrored packets inherit the forwarding class from the original packet (which is the forwarding class set by default classification that CoS applies to the packet on the ingress interface).

```
[edit services radius-flow-tap]
user@host# set forwarding-class class-name
```

5. (Optional) Specify the subscriber secure policy that determines what traffic, if any, is not sent to the mediation device.

```
[edit services radius-flow-tap]
user@host# set policy policy-name
```

NOTE: You can add or change a subscriber secure policy any time, but a changed policy does not apply to a currently enabled policy. To change a policy:

- Send a DTCP DELETE message to remove the current policy.
- Modify the configuration with the new version of the policy.
- Send a DTCP ADD message to add the policy.
- Send a DTCP ENABLE message to enable the policy.

6. (Optional) Specify the IP address for one or more target mediation devices to receive SNMPv3 trap notifications. Each target address must be configured separately.

```
[edit services radius-flow-tap]
user@host# set snmp notify-targets ip-address
```

> **NOTE**: You must also configure SNMP so that only encrypted notifications are sent to target devices. Targets without privacy configured cannot receive the notifications. For information about the SNMP configuration for subscriber secure policy, see "Configuring SNMPv3 Traps for Subscriber Secure Policy Mirroring" on page 678.

RELATED DOCUMENTATION

- Extensible Subscriber Services Manager Overview
- Subscriber Secure Policy Overview | 595
- Configuring RADIUS-Initiated Subscriber Secure Policy Mirroring Overview | 616
- Configuring DTCP-Initiated Subscriber Secure Policy Mirroring Overview | 636
- Guidelines for Configuring Subscriber Secure Policy Mirroring | 617

Configuring the Mediation Device as a User on the Router

In order for the router to receive DTCP messages from the mediation device, you need to configure the mediation device as a user on the router. To do so, create a login class that provides flow-tap operation permission and then create a login account that uses the login class.

To configure the mediation device as a user on the router:

1. Create the login class and configure `flow-tap-operation` permissions for the class.
   a. Specify that you want to configure login properties.

```
[edit system]
user@host# edit login
```

b. Create and name the class.

```
[edit system login]
user@host# edit class class-name
```

c. Configure the `flow-tap-operation` permission for the class.

```
[edit system login class class-name]
user@host# set permissions flow-tap-operation
```

2. Create the user login account for the mediation device.

a. Create the user account.

```
[edit system login]
user@host# edit user username
```

b. Configure the user ID.

```
[edit system login user username]
user@host# set uiduid-value
```

c. Configure the class for the user account.

```
[edit system login user username]
user@host# set class class-name
```

d. Configure the authentication for the user account.

```
[edit system login user username]
user@host# set authentication encrypted-password encrypted-password
```

Configuring a DTCP-over-SSH Connection to the Mediation Device

DTCP-initiated subscriber secure policy requires a DTCP-over-SSH connection for the radius-flow-tap service. This connection is used to send provisioning information from the mediation device to the router.

NOTE: DTCP-over-SSH connections are used for flow-tap, FlowTapLite, and radius-flow-tap services.

To configure the DTCP-over-SSH connection to support subscriber secure policy mirroring:

1. Access the `flow-tap-dtcp` hierarchy level.

```
[edit system services]
user@host# edit flow-tap-dtcp
```

NOTE:

2. Enable SSH support for DTCP.

```
[edit system services flow-tap-dtcp]
user@host# set ssh
```

3. (Optional) Configure the maximum number of established connections allowed for the DTCP service.

```
[edit system services flow-tap-service ssh]
user@host# set connection-limit limit
```

4. (Optional) Configure the maximum number of connection attempts allowed per minute for DTCP.

```
[edit system services flow-tap-service ssh]
user@host# set rate-limit limit
```

RELATED DOCUMENTATION

Configuring DTCP-Initiated Subscriber Secure Policy Mirroring Overview | 636

Configuring the Mediation Device to Provision Traffic Mirroring

To set up the mediation device to provision traffic mirroring on the router, use the following DTCP messages:

- To configure traffic-mirroring triggers, use the *ADD* message.
- To remove an existing traffic-mirroring trigger, use the *DELETE* message.
- To configure attributes to trigger a drop policy on the router (if one does not already exist), use the *ENABLE* message.
- To show existing traffic-mirroring triggers, use the *LIST* message.

NOTE: Consult the documentation for your mediation device to learn how to configure DTCP messages on the device.

For an example of how to use the DTCP messages, see "Example: Using DTCP Messages to Trigger, Verify, and Remove Traffic Mirroring for Subscribers" on page 665.

RELATED DOCUMENTATION

Configuring DTCP-Initiated Subscriber Secure Policy Mirroring Overview | 636

Disabling RADIUS-Initiated Subscriber Secure Policy Mirroring

DTCP-initiated and RADIUS-initiated subscriber secure policy mirroring both use the radius-flow-tap service. If you remove the `radius-flow-tap` configuration, then both types of mirroring are disabled. You can use the `dtcp-only` statement to cause RADIUS requests to initiate mirroring for a subscriber to be rejected; the mirroring service is not activated. The statement has no affect on DTCP-based mirroring.

Existing RADIUS-initiated mirroring is not affected by the statement, so to be effective you must issue the statement before a RADIUS-initiated service is activated for the subscriber. DTCP-initiated mirroring and FlowTapLite services, which use DTCP, are not affected.

To prevent RADIUS requests from initiating mirroring:

- Enable only DTCP support.

```
[edit system services]
user@host# set dtcp-only
```

RELATED DOCUMENTATION

Subscriber Secure Policy Overview | 595

Configuring Support for Subscriber Secure Policy Mirroring | 619

Configuring FlowTapLite on MX Series Routers and M320 Routers with FPCs

Example: Configuring Traffic That Is Mirrored Using DTCP-Initiated Subscriber Secure Policy

IN THIS SECTION

- Requirements | 645
- Overview | 645
- Configuration | 646

This example shows how to configure traffic that is mirrored using DTCP-initiated subscriber secure policy.

Requirements

- Juniper Networks MX Series routers.
- Junos OS Release 12.3R1 or later.

Overview

This example drops all video on demand TCP traffic from subnet 203.0.113.0/8 to any subscriber on which the policy named vod is enabled.

To configure traffic mirroring using DTCP-initiated subscriber secure policy:

1. Create a policy.
2. Set up the policy to filter IPv4 or IPv6 traffic by source or destination address, or port, protocol, or DSCP value.
3. Apply the policy using the DTCP attribute X-Drop-Policy.
4. Use the X-Drop-Policy with the DTCP ADD command to begin filtering traffic when mirroring is triggered.

NOTE: To begin filtering traffic that is currently being mirrored, use the X-Drop-Policy attribute with the DTCP ENABLE command. To stop filtering traffic that is currently being mirrored:

- Send a DTCP DELETE message to remove the current policy.
- Modify the configuration with the new version of the policy.
- Send a DTCP ADD message to add the policy.
- Send a DTCP ENABLE message to enable the policy.

Configuration

IN THIS SECTION

- Procedure | 646

Procedure

Step-by-Step Procedure

To configure filtering mirrored traffic before it is sent to a mediation device:

1. Specify that you want to configure radius-flow-tap.

```
[edit services]
user@host# edit radius-flow-tap
```

2. Specify that you want to configure a video on demand policy.

```
[edit services radius-flow-tap]
user@host# edit policy vod
```

3. Specify inet as the family that you want to use.

```
[edit services radius-flow-tap vod]
user@host# edit inet
```

4. Specify t1 as the term name for the IPv4 drop-policy.

```
[edit services radius-flow-tap vod inet]
user@host# edit drop-policy t1
```

5. Specify the source address for the drop-policy.

```
[edit services radius-flow-tap vod inet drop-policy t1]
user@host# edit source-address 203.0.113.0/8
```

6. Specify the match criteria that you want to use.

```
[edit services radius-flow-tap vod inet drop-policy t1]
user@host# set protocol tcp
```

Results

From configuration mode, confirm your configuration by entering the `show services` command. If the output does not display the intended configuration, repeat the instructions in this example to correct it.

```
[edit services radius-flow-tap policy]
vod {
    inet {
        drop-policy t1 {
            from{
                source-address {
                    203.0.113.0/8;
```

```
                }
                protocol tcp;
            }
        }
    }
```

If you are done configuring the device, enter `commit` from configuration mode.

RELATED DOCUMENTATION

- Subscriber Secure Policy Overview | **595**
- Configuring Support for Subscriber Secure Policy Mirroring | **619**

Terminating DTCP-Initiated Subscriber Traffic Mirroring Sessions

You can terminate DTCP-initiated traffic mirroring sessions by the following action:

- DTCP DELETE message receipt—Terminated upon receipt of a DTCP DELETE message. The DTCP administrator configures the DELETE message to include the same mirroring attributes that are used in the ADD message to initiate mirroring.

RELATED DOCUMENTATION

- *DELETE (DTCP)*
- DTCP Messages Used for Subscriber Secure Policy | **630**

CHAPTER 37

Configuring DTCP Messages Used for DTCP-Initiated Subscriber Secure Policy Mirroring

IN THIS CHAPTER

- ADD (DTCP) | 649
- DELETE (DTCP) | 655
- DISABLE (DTCP) | 657
- ENABLE (DTCP) | 659
- LIST (DTCP) | 662
- Example: Using DTCP Messages to Trigger, Verify, and Remove Traffic Mirroring for Subscribers | 665

ADD (DTCP)

IN THIS SECTION

- Syntax | 649
- Description | 650
- Options | 651
- Required Privilege Level | 654
- Sample Output | 654

Syntax

```
ADD DTCP/0.7
Csource-ID: user-name
Cdest-ID: variable
```

```
Priority: priority-number
X-Drop-Policy: policy-name
X-JTap-Cdest-Dest-Address: ipv4-address
X-JTap-Cdest-Dest-Port: udp-port
X-JTap-Cdest-Source-Address: ipv4-address
X-JTap-Cdest-Source-Port: port-number
X-JTap-Cdest-TTL: time-to-live
X-MD-Intercept-Id: 4-byte-id | 8-byte-id
Dtcp-trigger: trigger-value
Flags: flag
Seq: sequence-number
Authentication-Info: ssh-authentication-string
```

Description

Specify the DTCP attributes that do one of the following:

- Trigger the router to initiate traffic mirroring.
- Provide instructions to populate fields in the encapsulation header for packets sent to the mediation device

The DTCP ADD message can be sent either before or after subscribers log in through the interface.

The following attributes are added to the packet header of mirrored packets that the router sends to the mediation device. These attributes are required in the DTCP ADD message.

- X-JTap-Cdest-Dest-Address
- X-JTap-Cdest-Dest-Port
- X-MD-Intercept-Id

This DTCP message is supported for both FlowTapLite and radius-flow-tap services.

NOTE: Starting with Junos OS Release 12.3, DTCP ADD requests are validated for the IP version. The source IP and destination IP addresses must contain a matching IP address family, which must match with the value of the IPVersion field if it is available in the ADD message.

NOTE: Consult the documentation for your mediation device to learn how to configure DTCP messages on the device.

BEST PRACTICE: The Account Session ID, Interface Identifier, and Subscriber User Name trigger attributes are optimized for a scaled subscriber management environment. Forwarding of mirrored traffic begins almost immediately when you include one or more of these three attributes and none of the non-optimized attributes in DTCP ADD messages.

If you include any of the non-optimized trigger attributes in the DTCP ADD message in a scaled subscriber management environment, some delay might be observed between the time when the DTCP ADD message is sent and the time when forwarding starts for the mirrored traffic. For example, if there are 10,000 subscriber sessions on the router, forwarding of the mirrored traffic might be delayed for less than one minute. This delay occurs when you specify any non-optimized attribute, with or without any optimized attribute. The delay occurs regardless of the order of attributes in the DTCP packet.

When a subscriber matches more than one of the DTCP mirroring triggers in an ADD message, the router processes the triggers in the following order:

1. `X-Act-Sess-Id`
2. `X-Call-Sta-Id`
3. `X-IP-Addr`
4. `X-Interface-Id`
5. `X-NAS-Port-Id`
6. `X-RM-Circuit-Id`
7. `X-UserName`

BEST PRACTICE: When you have DHCPv4/DHCPv6 subscribers over VLANs, two sessions are created for each subscriber— one for the Layer 2 VLAN, and one for DHCP. In this case do not use a trigger, such as `X-RM-Circuit-Id`, that applies to both the VLAN and the DHCP sessions. If the DHCP and VLAN sessions match the same trigger, the DHCP subscriber login fails and subscriber secure policy is not triggered. You need to select a traffic mirroring trigger that matches only one of these sessions.

Options

Csource-ID: ***username***

Username on the router. This username must be configured as a DTCP user on the router using the `set system login class` or `set system login user` statements.

Cdest-ID: *variable*	ID of the mediation device.
Flags: *flag*	STATIC is the only flag supported.
Priority: *priority-number*	This implementation of DTCP does not use the priority number.
X-Drop-Policy *policy-name*	Name of the policy used to determine which mirrored packets are no longer sent to the mediation device.
X-JTap-Cdest-Dest-Address: *ipv4-address*	Destination IPv4 address of the mediation device to which intercepted packets are sent. You must include this attribute in your ADD messages. It is used in the header of mirrored traffic that is sent to the mediation device.
X-JTap-Cdest-Dest-Port: *udp-port*	Destination port of the mediation device to which intercepted packets are sent. You must include this attribute in your ADD messages. It is used in the header of mirrored traffic that is sent to the mediation device.
X-JTap-Cdest-Source-Address: *ipv4-address*	Source IPv4 address. You must include this attribute in your ADD messages. If the value entered does not match the value configured on the router using the `set services radius-flow-tap source-ipv4-address` *`source-ipv4-address`* statement, it is replaced by configured value.
X-JTap-Cdest-Source-Port: *port-number*	Source port. You must include this attribute in your ADD messages. If the value entered does not match the value of X-Jtap-Cdest-Dest-Port, it is ignored.
X-JTap-Cdest-TTL: *time-to-live*	TTL value to be used in the forwarded packet.
X-MD-Intercept-Id *4-byte-id* **or** *8-byte-id*	An Id that is used to identify a subscriber. You must include this attribute in your ADD messages. This ID is used in the header of mirrored traffic that is sent to the mediation device to allow the device to track a subscriber. The X-MD-Intercept-ID attribute must be provided in hexadecimal format, it can be prepended with 0x or 0X, but this prepend is optional. The X-MD-Intercept-ID attribute can consist of only 4 bytes or 8 bytes. If 4 bytes format is used, the two most significant bits must be 01. If 8 bytes format is used, the two most significant bits must be 00.
Dtcp-trigger*: *trigger-value	DTCP attribute used to trigger traffic mirroring. • `X-Act-Sess-Id`—Text string of the accounting session ID associated with the subscriber session. The intercept terminates when the subscriber logs out.

BEST PRACTICE: We recommend that you include other triggers to ensure that all sessions for a subscriber are intercepted.

- `X-Call-Sta-Id`—Text string of the calling station ID associated with the subscriber. If the subscriber is not logged in, the policy is applied at any current or subsequent subscriber log in.
- `X-IP-Addr`—IPv4 address that is associated with the interface for a subscriber.

  If the subscriber is not using the default logical system, you must also include the `X-Logical-System` attribute in your DTCP message. If the subscriber is not using the default routing instance, you must also include the `X-Router-Instance` attribute in your DTCP message.
- `X-Interface-Id`—Interface description string on which traffic mirroring is performed. Traffic is mirrored for all subscribers that use this interface; for example, `ge-0/0/0.1` or `demux0.107472834`.
- `X-NAS-Port-Id`—Text string of the NAS port ID associated with the subscriber.
- `X-RM-Circuit-Id`—For PPPoE subscribers, the agent circuit ID (ACI) in the PPPoE Intermediate Agent (PPPoE IA) tag.

  For DHCP subscribers, use `X-RM-Circuit-Id` with the agent remote ID (ARI), `X-RM-Agent-Id`, to completely specify a trigger for the DHCP option 82 value that is associated with this session.
- `X-RM-Agent-Id`—For PPPoE subscribers, the agent remote ID (ARI) in the PPPoE IA tag.

  For DHCP subscribers, `X-RM-Agent-Id` is the option 82 Agent-Remote-ID suboption and you can use it alone as a trigger. You can also use it with the ACI, `X-RM-Circuit-Id`, to completely specify a trigger for the DHCP option 82 value that is associated with this session.
- `X-Logical-System`—Include in addition to the `X-IP-Addr` or `X-UserName` attribute for subscribers that use anything other than the default logical system. `X-Logical-System` is ignored if neither of those attributes is included in the message. The default logical system is assumed when `X-Logical-System` is not included in the ADD message.

- X-Router-Instance—Include in addition to the X-IP-Addr or X-UserName attribute for subscribers that use anything other than the default routing instance. X-Router-Instance is ignored if neither of those attributes is included in the message. The default routing instance is assumed when X-Router-Instance is not included in the ADD message.
- X-UserName—Subscriber's user name. For subscribers not using the default logical system or routing instance, you can also include the X-Logical-System or X-Router-Instance attributes.

Seq: ***sequence-number*** Number added by the mediation device. DTCP messages contain a monotonically increasing sequence number for each successive message.

Authentication-Info: ***ssh-authentication-string*** String used when you are using SSH to connect to the router.

Required Privilege Level

Not applicable.

Sample Output

command-name

```
ADD DTCP/0.7
Csource-ID: ft-user1
Cdest-ID: cd1
Priority: 2
X-JTap-Cdest-Dest-Address: 203.0.113.50
X-JTap-Cdest-Dest-Port: 7890
X-JTap-Cdest-Source-Address: 203.0.113.9
X-JTap-Cdest-Source-Port: 12321
X-Interface-Id: ge-0/0/2.1
X-MD-Intercept-Id: 55667788
Flags: STATIC
Seq: 1
Authentication-Info: c16d2d9d1679facf0c4a66683af6114d341e4033
DTCP/0.7 200 OK
SEQ: 7
```

```
CRITERIA-ID: 2
TIMESTAMP: 2011-02-13 15:56:49.609
```

RELATED DOCUMENTATION

- DTCP Messages Used for Subscriber Secure Policy | 630
- Packet Header for Mirrored Traffic Sent to Mediation Device | 631
- DTCP-Initiated Subscriber Secure Policy Overview | 624
- Example: Using DTCP Messages to Trigger, Verify, and Remove Traffic Mirroring for Subscribers | 665

DELETE (DTCP)

IN THIS SECTION

- Syntax | 655
- Description | 656
- Options | 656
- Required Privilege Level | 656
- Sample Output | 656

Syntax

```
DELETE DTCP/0.7
Csource-ID: user-name
CRITERIA-ID: criteria-id
Cdest-ID: variable
Flags: flag
Seq: sequence-number
Authentication-Info:ssh-authentication-string
```

Description

Remove traffic mirroring for a subscriber. Mirroring of the existing subscriber is stopped. This DTCP message is supported for both FlowTapLite and radius-flow-tap services.

NOTE: Consult the documentation for your mediation device to learn how to configure DTCP messages on the device.

Options

Csource-ID: *user-name*	Username on the router. This name must be configured on the router.
CRITERIA-ID: *criteria-id*	ID that DTCP assigns for the mirrored session when you create a DTCP ADD message. Use this ID in your DELETE messages to remove the intercept for a specific subscriber. To view the ID, use the DTCP LIST message. The CRITERIA-ID and the Cdest-ID are mutually exclusive in DELETE messages.
Cdest-ID: *variable*	ID of the mediation device. Use this ID in your DELETE messages to remove all mirroring sessions associated with a mediation device. The Cdest-ID and the CRITERIA-ID are mutually exclusive in DELETE messages.
Flags: *flag*	STATIC is the only flag supported.
Seq: *sequence-number*	Number added by the mediation device. DTCP messages contain a monotonically increasing sequence number for each successive message.
Authentication-Info: *ssh-authentication-string*	String used when you are using SSH to connect to the router.

Required Privilege Level

Not applicable.

Sample Output

The following sample shows how to remove mirroring for a specific subscriber by using the CRITERIA-ID.

DELETE DTCP

```
DELETE DTCP/0.7
Csource-ID: dtcp1
CRITERIA-ID: 2
Flags: STATIC
Seq: 10
Authentication-Info: 7e84ae871b12f2da023b038774115bb8d955f17e
DTCP/0.7 200 OK
SEQ: 10
CRITERIA-COUNT: 1
TIMESTAMP: 2011-02-13 16:00:02.802
AUTHENTICATION-INFO: 2834ff32ec07d84753a046cfb552e072cc27d50b
```

RELATED DOCUMENTATION

DTCP Messages Used for Subscriber Secure Policy | 630

DTCP-Initiated Subscriber Secure Policy Overview | 624

DISABLE (DTCP)

IN THIS SECTION

- Syntax | 658
- Description | 658
- Options | 658
- Required Privilege Level | 658
- Sample Output | 659
- Release Information | 659

Syntax

```
DISABLE DTCP/0.8
Csource-ID: user-name
Criteria-ID: variable
X-Drop-Policy: variable
Flags: flags
```

Description

Specify the DTCP attributes used in DISABLE messages to remove a drop policy created by a prior DTCP ENABLE command.

This DTCP message is supported for the radius-flow-tap service. It is not supported for the FlowTapLite service.

NOTE: Consult the documentation for your mediation device to learn how to configure DTCP messages on the device.

Options

Csource-ID: *user-name*	Username on the router. This username must be configured as a DTCP user on the router using the `set system login class` or `set system login user` statements.
Criteria-ID: *variable*	Value returned from a prior DTCP ADD that identifies the trigger on which to disable this drop policy.
Flags: *flag*	STATIC is the only flag supported.
X-Drop-Policy: *variable*	Name of the policy that determines which mirrored packets are no longer sent to the mediation device.

Required Privilege Level

Not applicable.

Sample Output

command-name

```
ENABLE DTCP/0.8DISABLE DTCP/0.8
Csource-ID: ft-user
Criteria-ID: 8
X-Drop-Policy: drop_pol
Flags: STATIC
Seq: 1

DTCP/0.8 200 OK
SEQ: 1
CRITERIA-COUNT: 1
TIMESTAMP: 2022-01-24 11:58:47.927
```

Release Information

Command introduced in Junos OS Release 12.3.

RELATED DOCUMENTATION

DTCP Messages Used for Subscriber Secure Policy | 630

DTCP-Initiated Subscriber Secure Policy Overview | 624

ENABLE (DTCP)

IN THIS SECTION

- Syntax | 660
- Description | 660
- Options | 660
- Required Privilege Level | 661
- Sample Output | 661

Release Information | 661

Syntax

```
ENABLE DTCP/0.8
Csource-ID: user-name
Criteria-ID: variable
X-Drop-Policy: variable
Flags: flags
```

Description

Specify the DTCP attributes used in ENABLE messages to cause the router to trigger a drop policy if one does not already exist from a prior DTCP ADD or DTCP ENABLE command.

The DTCP ENABLE message can only be issued on a Criteria-ID that was returned in a response to a previous DTCP ADD command. The policy applies to any new subscribers who match the trigger corresponding to the Criteria-ID. Any existing mirroring remains in place and the policy is not be applied to them. The DTCP ENABLE command stops only the traffic that is identified by the specified policy from being sent to the mediation device.

This DTCP message is supported for the radius-flow-tap service. It is not supported for the FlowTapLite service.

NOTE: Consult the documentation for your mediation device to learn how to configure DTCP messages on the device.

Options

Csource-ID: ***user-name***	Username on the router. This username must be configured as a DTCP user on the router using the `set system login class` or `set system login user` statements.
Criteria-ID: ***variable***	Value returned from a prior DTCP ADD that identifies the trigger on which to disable this drop policy.
Flags: ***flag***	STATIC is the only flag supported.

X-Drop-Policy: *variable* — Name of the policy that determines which mirrored packets are no longer sent to the mediation device.

Required Privilege Level

Not applicable.

Sample Output

command-name

```
ENABLE DTCP/0.8
Csource-ID: ft-user1
Criteria-ID: 1
X-Drop: T1
Flags: STATIC
Seq: 1
Authentication-Info: c16d2d9d1679facf0c4a66683af6114d341e4033

DTCP/0.8 200 OK
SEQ: 7
CRITERIA-ID: 2
TIMESTAMP: 2011-02-13 15:56:49.609
```

Release Information

Command introduced in Junos OS Release 12.3.

RELATED DOCUMENTATION

DTCP Messages Used for Subscriber Secure Policy | 630

DTCP-Initiated Subscriber Secure Policy Overview | 624

LIST (DTCP)

IN THIS SECTION

- Syntax | 662
- Description | 662
- Options | 663
- Required Privilege Level | 663
- Sample Output | 663

Syntax

```
LIST DTCP/0.7
Csource-ID: user-name
Cdest-ID: variable
Flags: BOTH
Seq: sequence-number
Authentication-Info: ssh-authentication-string
```

Description

Request information that is returned in a LIST response. The response lists triggers only. It does not return sessions that are being mirrored. This DTCP message is supported for both FlowTapLite and radius-flow-tap services.

NOTE: Consult the documentation for your mediation device to learn how to configure DTCP messages on the device.

NOTE: You can ignore the following fields in command output: AVERAGE-BANDWIDTH, MATCHING-PACKETS, MATCHING-BYTES, and NUM-REFRESH. The LAST-REFRESH field shows the time when the corresponding DTCP ADD request was processed.

Options

Csource-ID: *username*
Username on the router. This name must be configured on the router.

Cdest-ID: *variable*
ID of the mediation device.

If a LIST DTCP command is sent with multiple Cdest-IDs, the error `400 Bad Request` is displayed.

Flags: *flag*
BOTH is the only flag supported. This field must be included in the LIST message for the LIST request to not be rejected until Junos OS Releases 14.1R4 and 14.2R2. Starting with Junos OS Release 14.1R5, 14.2R3, and 15.1R1, the Flags field is not a required parameter in the DTCP LIST message. The LIST request is not rejected if the LIST message does not contain the Flags field. If the DTCP LIST message contains the Flags field, the value of that field is processed. If the LIST message does not contain the Flags field, the CRITERIA field parameter is used for the Flags field.

Starting with Junos OS Release 12.3, when more than one CDest-ID parameter is present in the DTCP LIST or DELETE DTCP commands, the error code 400 (Bad Request) is returned in the response, instead of the error code 431 (Unknown Content Destination).

Seq: *sequence-number*
Number added by the mediation device. DTCP messages contain a monotonically increasing sequence number for each successive message.

Authentication-Info: *ssh-authentication-string*
String used when you are using SSH to connect to the router.

Required Privilege Level

Not applicable.

Sample Output

LIST DTCP

```
LIST DTCP/0.7
Csource-ID: dtcp1
Cdest-ID: cd1
Flags: BOTH
Seq: 9
```

```
Authentication-Info: f6dd64643021debb167ce2fb2d3c7b6622a87e09
DTCP/0.7 200 OK
SEQ: 9
TIMESTAMP: 2011-02-13 15:57:47.667
CRITERIA-ID: 2
CSOURCE-ID: dtcp1
CDEST-ID: cd1
CSOURCE-ADDRESS: 203.0.113.224
FLAGS: BOTH
AVERAGE-BANDWIDTH: 0
MATCHING-PACKETS: 0
MATCHING-BYTES: 0
NUM-REFRESH: 0
LAST-REFRESH: 2019-06-13 23:45:34.734
X-JTAP-CDEST-DEST-ADDRESS: 192.0.2.168
X-JTAP-CDEST-DEST-PORT: 65535
X-JTAP-CDEST-SOURCE-ADDRESS: 198.51.100.10
X-JTAP-CDEST-SOURCE-PORT: 50000
X-JTAP-CDEST-TTL: 64
X-INTERFACE-ID: demux0.30010002
X-MD-INTERCEPT-ID: 0x0101010130010002
CRITERIA-NUM: 1
CRITERIA-COUNT: 0
CRITERIA-ID: 3
CSOURCE-ID: dtcp1
CDEST-ID: cd1
CSOURCE-ADDRESS: 203.0.113.224
FLAGS: BOTH
AVERAGE-BANDWIDTH: 0
MATCHING-PACKETS: 0
MATCHING-BYTES: 0
NUM-REFRESH: 0
LAST-REFRESH: 2019-06-13 23:45:48.912
X-JTAP-CDEST-DEST-ADDRESS: 192.0.2.168
X-JTAP-CDEST-DEST-PORT: 65535
X-JTAP-CDEST-SOURCE-ADDRESS: 198.51.100.10
X-JTAP-CDEST-SOURCE-PORT: 50000
X-JTAP-CDEST-TTL: 64
X-INTERFACE-ID: demux0.30010001
X-MD-INTERCEPT-ID: 0x0101010130010001
CRITERIA-NUM: 2
CRITERIA-COUNT: 2
```

```
AUTHENTICATION-INFO: 361171ccb24dde6afe8ef66021287f9b8ac16028
```

RELATED DOCUMENTATION

DTCP Messages Used for Subscriber Secure Policy | 630

DTCP-Initiated Subscriber Secure Policy Overview | 624

Example: Using DTCP Messages to Trigger, Verify, and Remove Traffic Mirroring for Subscribers

IN THIS SECTION

- Creating DTCP ENABLE Messages to Trigger a Drop Policy | 666
- Creating DTCP DISABLE Messages to Remove a Drop Policy | 666
- Creating DTCP ADD Messages to Trigger Traffic Mirroring | 667
- Creating DTCP ENABLE Messages to Trigger Traffic Mirroring | 668
- Using LIST Messages to Verify That Subscriber Traffic Is Being Mirrored | 669
- Using DELETE Messages to Remove Traffic Mirroring Triggers | 670
- Verifying That Traffic Mirroring Was Stopped on the Subscriber Interfaces | 671

This example shows how to create DTCP messages to do the following:

- Trigger a drop policy if one does not already exist.
- Remove an existing drop policy.
- Trigger traffic mirroring for two subscribers based on interface ID.
- Verify that subscriber traffic on the two interfaces is being mirrored.
- Remove traffic mirroring on the two subscriber interfaces.
- Verify that traffic mirroring was stopped on the two subscriber interfaces.

In this example, SSH is being used to communicate with the router.

Creating DTCP ENABLE Messages to Trigger a Drop Policy

This section shows the DTCP attributes used in ENABLE messages to cause the router to trigger a drop policy if one does not already exist from a prior DTCP ADD or DTCP ENABLE command.

```
ENABLE DTCP/0.7
Csource-ID: ft-user1
Criteria-ID: 1
X-Drop: T1
Flags: STATIC
Seq: 1
Authentication-Info: c16d2d9d1679facf0c4a66683af6114d341e4033

DTCP/0.7 200 OK
SEQ: 7
CRITERIA-ID: 2
TIMESTAMP: 2011-02-13 15:56:49.609
```

Creating DTCP DISABLE Messages to Remove a Drop Policy

This section shows the DTCP attributes used in DISABLE messages to cause the router to remove an existing a drop policy created with a prior DTCP ENABLE command.

```
DISABLE DTCP/0.8
Csource-ID: ft-user
Criteria-ID: 8
X-Drop-Policy: drop_pol
Flags: STATIC
Seq: 1
Authentication-Info: 963aa01c38c531f63fb410afd058c018be4d0230


DTCP/0.8 200 OK
SEQ: 1
CRITERIA-COUNT: 1
TIMESTAMP: 2022-01-24 11:58:47.927
```

Creating DTCP ADD Messages to Trigger Traffic Mirroring

This section shows examples of DTCP ADD messages on a mediation device that use the interface ID to trigger traffic mirroring on interfaces demux0.30010002 and demux0.30010001.

```
ADD DTCP/0.7
Csource-ID: dtcp1
Cdest-ID: cd1
Priority: 2
X-JTap-Cdest-Dest-Address: 192.0.2.168
X-JTap-Cdest-Dest-Port: 65535
X-JTap-Cdest-Source-Address: 198.51.100.10
X-JTap-Cdest-Source-Port: 50000
X-JTap-Cdest-TTL: 64
X-Interface-Id: demux0.30010002         /*Used as trigger*/
X-MD-Intercept-Id: 0x0101010130010002
Flags: BOTH
Seq: 7
Authentication-Info: c16d2d9d1679facf0c4a66683af6114d341e4033


DTCP/0.7 200 OK
SEQ: 7
CRITERIA-ID: 2
TIMESTAMP: 2011-02-13 15:56:49.609
AUTHENTICATION-INFO: 4880de4b8cead98c95813fd9b95e240b107d4693
```

```
ADD DTCP/0.7
Csource-ID: dtcp1
Cdest-ID: cd1
Priority: 2
X-JTap-Cdest-Dest-Address: 192.0.2.168
X-JTap-Cdest-Dest-Port: 65535
X-JTap-Cdest-Source-Address: 198.51.100.10
X-JTap-Cdest-Source-Port: 50000
X-JTap-Cdest-TTL: 64
X-Interface-Id: demux0.30010001         /*Used as trigger*/
X-MD-Intercept-Id: 0x0101010130010001
Flags: STATIC
Seq: 8
```

```
Authentication-Info: dc3c55481a3810c7dd29fdc1b4681d978ff4e7c4

DTCP/0.7 200 OK
SEQ: 8
CRITERIA-ID: 3
TIMESTAMP: 2011-02-13 15:57:20.640
AUTHENTICATION-INFO: 4b31ef1311647e5ba52d2d5d4237b9e5beaa47b7
```

```
ADD DTCP/0.7
Csource-ID: ft-user1
Cdest-ID: cd1
Priority: 2
X-JTap-Cdest-Dest-Address:  203.0.113.112
X-JTap-Cdest-Dest-Port: 7899
X-JTap-Cdest-Source-Address: 192.0.2.9
X-JTap-Cdest-Source-Port: 12321
X-Username: testuser
X-MD-Intercept-Id: 55667789
Flags: STATIC


DTCP/0.7 200 OK
SEQ: 100
CRITERIA-ID: 1
```

Creating DTCP ENABLE Messages to Trigger Traffic Mirroring

This section shows the DTCP attributes used in ENABLE messages to cause the router to trigger a drop policy if one does not already exist from a prior DTCP ADD or DTCP ENABLE command.

```
ENABLE DTCP/0.8
Csource-ID: ft-user1
Cdest-ID: cd1
X-Drop-Policy: vod
Flags: STATIC
```

Using LIST Messages to Verify That Subscriber Traffic Is Being Mirrored

This section shows examples of a LIST message on the mediation device. The LIST message requests information about the subscribers being mirrored. The information is returned in a LIST response. The response shows that traffic for the two interfaces—demux0.30010002 and demux0.30010001—is being mirrored.

```
LIST DTCP/0.7
Csource-ID: dtcp1
Cdest-ID: cd1
Seq: 9
Authentication-Info: f6dd64643021debb167ce2fb2d3c7b6622a87e09


DTCP/0.7 200 OK
SEQ: 9
TIMESTAMP: 2011-02-13 15:57:47.667
CRITERIA-ID: 2
CSOURCE-ID: dtcp1
CDEST-ID: cd1
CSOURCE-ADDRESS: 203.0.113.224
FLAGS: BOTH
X-JTAP-CDEST-DEST-ADDRESS: 192.0.2.168
X-JTAP-CDEST-DEST-PORT: 65535
X-JTAP-CDEST-SOURCE-ADDRESS: 198.51.100.10
X-JTAP-CDEST-SOURCE-PORT: 50000
X-JTAP-CDEST-TTL: 64
X-INTERFACE-ID: demux0.30010002        /*subscriber interface*/
X-MD-INTERCEPT-ID: 0x0101010130010002
CRITERIA-NUM: 1
CRITERIA-COUNT: 0

CRITERIA-ID: 3
CSOURCE-ID: dtcp1
CDEST-ID: cd1
CSOURCE-ADDRESS: 203.0.113.224
FLAGS: BOTH
X-JTAP-CDEST-DEST-ADDRESS: 192.0.2.168
X-JTAP-CDEST-DEST-PORT: 65535
X-JTAP-CDEST-SOURCE-ADDRESS: 198.51.100.10
X-JTAP-CDEST-SOURCE-PORT: 50000
X-JTAP-CDEST-TTL: 64
```

```
X-INTERFACE-ID: demux0.30010001     /*subscriber interface*/
X-MD-INTERCEPT-ID: 0x0101010130010001
CRITERIA-NUM: 2
CRITERIA-COUNT: 2
AUTHENTICATION-INFO: 361171ccb24dde6afe8ef66021287f9b8ac16028
```

Using DELETE Messages to Remove Traffic Mirroring Triggers

This section shows examples of DELETE messages used to remove traffic mirroring triggers on demux0.30010001 and demux0.30010002. DTCP DELETE can use either Criteria-ID to delete only that criteria or Cdest-ID to delete everything with cdest-ID that you previously created.

```
DELETE DTCP/0.7
Csource-ID: dtcp1
CRITERIA-ID: 2
Flags: STATIC
Seq: 10
Authentication-Info: 7e84ae871b12f2da023b038774115bb8d955f17e


DTCP/0.7 200 OK
SEQ: 10
CRITERIA-COUNT: 1
TIMESTAMP: 2011-02-13 16:00:02.802
AUTHENTICATION-INFO: 2834ff32ec07d84753a046cfb552e072cc27d50b


DELETE DTCP/0.7
Csource-ID: dtcp1
CRITERIA-ID: 3
Flags: STATIC
Seq: 12
Authentication-Info: 7653fd94659a7183a990bdea654a1b97c0895348

DTCP/0.7 200 OK
SEQ: 12
CRITERIA-COUNT: 1
TIMESTAMP: 2011-02-13 16:01:35.895
AUTHENTICATION-INFO: 7cd8171057a327434e1b2d9b35f43b88305f9a74
```

Verifying That Traffic Mirroring Was Stopped on the Subscriber Interfaces

This section shows an example of a LIST message used to show that traffic mirroring on demux0.30010001 and demux0.30010002 is removed.

```
LIST DTCP/0.7
Csource-ID: dtcp1
Cdest-ID: cd1
Seq: 13
Authentication-Info: 7c9f825427cfeaecebb0d13ea3842af1021c7d26

DTCP/0.7 430 Unknown Content Destination
SEQ: 13
AUTHENTICATION-INFO: 5ca2eec65106354fe59c878b4c36b7de3c511acd
```

RELATED DOCUMENTATION

DTCP-Initiated Subscriber Secure Policy Overview | **624**

Configuring DTCP-Initiated Subscriber Secure Policy Mirroring Overview | **636**

CHAPTER 38

Configuring Subscriber Secure Policy Support for IPv4 Multicast Traffic

IN THIS CHAPTER

- Subscriber Secure Policy Support for IPv4 Multicast Traffic | 672
- Enabling Subscriber Secure Policy Mirroring for IPv4 Multicast Traffic | 673

Subscriber Secure Policy Support for IPv4 Multicast Traffic

IN THIS SECTION

- Triggering the Mirroring of IPv4 Multicast Traffic | 672

IP multicast traffic is used for applications such as audio or video streaming, IPTV, video conferencing, or online gaming. Multicast traffic is sent to multiple subscribers who have joined a multicast group.

Secure subscriber policy allows for the mirroring of IPv4 multicast traffic sent to a specific subscriber. If multiple subscribers whose traffic requires mirroring join the same multicast session, the subscriber secure policy feature mirrors each subscriber's traffic and forwards it separately to the mediation device with the proper prepended header.

Mirroring of multicast traffic is supported only for subscribers in the default logical system.

You can enable and disable the mirroring of multicast traffic on a per-chassis basis. You cannot enable or disable it on a per-subscriber basis.

Triggering the Mirroring of IPv4 Multicast Traffic

Multicast traffic being sent towards a subscriber does not contain much of the identifying information used to trigger mirroring of a subscriber's unicast traffic. For example, the multicast packet contains the

multicast group address in the destination address of the packet instead of the subscriber's IP address. It also does not contain the user name or MAC address of the subscriber, and does not include information obtained by RADIUS or DHCP. Therefore, methods of identifying multicast traffic that is received by a subscriber are not the same as methods of identifying a subscriber's unicast traffic or multicast traffic that is sent by a subscriber.

To join a multicast group, a subscriber sends an IGMP join request, and it receives a reply. The reply contains the multicast groups to which the subscriber is registered. Triggering the mirroring of multicast traffic is based on the sending of the IGMP join request and the information in the IGMP reply. If the subscriber's unicast traffic is already being mirrored either through DTCP-initiated or RADIUS-initiated traffic mirroring, and the subscriber sends an IGMP join request, mirroring of multicast traffic sent to the subscriber is initiated. The traffic being mirrored is based on the groups contained in the IGMP reply.

RELATED DOCUMENTATION

Enabling Subscriber Secure Policy Mirroring for IPv4 Multicast Traffic | 673

Enabling Subscriber Secure Policy Mirroring for IPv4 Multicast Traffic

This topic describes the steps to enable subscriber secure policy mirroring of IPv4 multicast traffic. You can enable and disable IPv4 multicast intercept on a per chassis basis.

To configure subscriber secure policy to support IPv4 multicast traffic mirroring:

1. Configure the radius-flow-tap service used for subscriber secure policy mirroring.

```
[edit services]
user@host# edit radius-flow-tap
```

2. Enable the interception of multicast traffic.

```
[edit services radius-flow-tap]
user@host# set multicast-interception
```

RELATED DOCUMENTATION

Subscriber Secure Policy Support for IPv4 Multicast Traffic | 672

Configuring RADIUS-Initiated Subscriber Secure Policy Mirroring Overview | 616

Configuring DTCP-Initiated Subscriber Secure Policy Mirroring Overview | 636

CHAPTER 39

Configuring Intercept-Related Information for Subscriber Secure Policy

IN THIS CHAPTER

- Intercept-Related Events Transmitted to the Mediation Device | 675
- SNMP Traps for Subscriber Secure Policy LAES Compliance | 676
- Configuring SNMPv3 Traps for Subscriber Secure Policy Mirroring | 678
- Example: SNMPv3 Traps Configuration for Subscriber Secure Policy Mirroring | 679

Intercept-Related Events Transmitted to the Mediation Device

You can use SNMPv3 traps to report intercept-related events to the mediation device. These events include identifying information for subscribers, such as username or IP address, and subscriber session events, such as login or logout events or mirroring session activation or deactivation. The router sends the events to the mediation device in SNMP traps. Using SNMPv3 with privacy (encryption) configured provides secure traps that are visible only to authorized individuals on the intended secure mediation device. The traps help support compliance with the Communications Assistance for Law Enforcement Act (CALEA), which defines electronic surveillance guidelines for telecommunications companies.

The supported SNMPv3 traps map to messages defined by the *Lawfully Authorized Electronic Surveillance (LAES) for IP Network Access, American Nation Standard For Telecommunications*. "SNMP Traps for Subscriber Secure Policy LAES Compliance" on page 676 describes the supported SNMPv3 traps and their related LAES messages.

RELATED DOCUMENTATION

- Subscriber Secure Policy Overview | 595
- Configuring RADIUS-Initiated Subscriber Secure Policy Mirroring Overview | 616
- Configuring DTCP-Initiated Subscriber Secure Policy Mirroring Overview | 636
- SNMP Traps for Subscriber Secure Policy LAES Compliance | 676

SNMP Traps for Subscriber Secure Policy LAES Compliance

Table 48 on page 676 describes the SNMPv3 traps that subscriber secure policy mirroring uses to provide information that maps to messages defined in the *Lawfully Authorized Electronic Surveillance (LAES) for IP Network Access, American National Standard for Telecommunications*. These messages enable subscriber secure policy to comply with the *Communications Assistance for Law Enforcement Act* (CALEA). The Juniper Packet Mirroring MIB, **jnx-js-packet-mirror.mib**, provides the SNMP trap.

Table 48: Subscriber Secure Policy SNMPv3 Traps for LAES Messages

SNMPv3 Trap	LAES Message	Description
jnxJsPacketMirrorLiSubscriberLoggedIn	• access-attempt (implied) • access-session-accept • packet-data-session-start	A subscriber, who is identified to have a mirrored service that is activated at login, has successfully logged in.
jnxJsPacketMirrorSessionLiSubscriberLogInFailed	• access-attempt (implied) • access-failed (all termination reasons except authentication-reject) • access-reject (termination reason is authentication-reject)	A subscriber, who is identified to have a mirrored service that is activated at login, has failed to log in.
jnxJsPacketMirrorInterfaceLiSubscriberLoggedOut	• access-session-end • packet-data-session-end	A subscriber, who had an active mirrored service, has logged out.
jnxJsPacketMirrorInterfaceLiServiceActivated	• packet-data-session-already-established	A mirrored session has been activated.
jnxJsPacketMirrorSessionLiServiceActivationFailed	–	A mirrored session for a subscriber has failed.

Table 48: Subscriber Secure Policy SNMPv3 Traps for LAES Messages *(Continued)*

SNMPv3 Trap	LAES Message	Description
jnxJsPacketMirrorSessionLiServiceDeactivated	–	A mirrored session for an established subscriber has been deactivated.
jnxJsPacketMirrorMirroringFailure	–	A mirrored service request failed due to an invalid value in the request. Note: This trap is not related to LAES messages.
jnxJsPacketMirrorTriggerType	–	The type of trigger that caused the mirroring session to be activated.
jnxJsPacketMirrorCallingStationIdentifier	–	The calling station ID of the subscriber whose traffic is currently being mirrored.
jnxJsPacketMirrorNasIdentifier	–	The NAS ID of the session in which traffic is being mirrored.
jnxJsPacketMirrorTargetIPv6Address	–	The IPv6 address of the subscriber interface that is being mirrored.

RELATED DOCUMENTATION

Intercept-Related Events Transmitted to the Mediation Device | 675

Example: SNMPv3 Traps Configuration for Subscriber Secure Policy Mirroring | 679

Configuring SNMPv3 Traps for Subscriber Secure Policy Mirroring

This topic provides an overview of the SNMPv3 configuration process as it pertains to subscriber secure policy.

To configure SNMPv3 trap support for subscriber secure policy and to send the trap information to the mediation device:

1. Configure the MIB view.

   See Configure MIB Views.
2. Configure the trap notification and trap notification filter. See the following topics:
   - Configure SNMPv3 Trap Notification
   - Configuring the Trap Notification Filter
3. Configure the target device. The target device is the mediation device that receives the trap information.

   See Configuring SNMPv3 Traps on a Device Running Junos OS.

   > **NOTE**: Starting in Junos OS Release 16.1R1, when you configure SNMP trap notifications for subscriber secure policy on MX Series routers, you must configure the target parameters for mediation devices so that the SNMPv3 traps are sent with privacy (encrypted). Targets without privacy configured cannot receive the notifications. In earlier releases, you can configure target parameters without privacy, allowing unencrypted notifications to be sent to the mediation devices.
   >
   > For more information about configuring subscriber secure policies, see "Subscriber Secure Policy Overview" on page 595.
4. Configure the SNMPv3 user, authentication method and password, and privacy method and password. See the following topics:
   - Creating SNMPv3 Users
   - Configuring the SNMPv3 Authentication Type
   - Configuring the SNMPv3 Encryption Type
5. Configure user access privileges to management information.

   See Defining Access Privileges for an SNMP Group.

RELATED DOCUMENTATION

Intercept-Related Events Transmitted to the Mediation Device | 675

SNMP Traps for Subscriber Secure Policy LAES Compliance | 676

Example: SNMPv3 Traps Configuration for Subscriber Secure Policy Mirroring | 679

Configure SNMPv3

Example: SNMPv3 Traps Configuration for Subscriber Secure Policy Mirroring

This example shows an SNMP configuration that provides SNMPv3 trap support.

Configure the SNMPv3 trap support.

```
[edit snmp]
v3 {
    usm {
        local-engine {
            user mediation-device1 { ## Name of the mediation device
                authentication-md5 {
                    authentication-key "$ABC123$ABC123"; ## SECRET-DATA
                }
                privacy-des {
                    privacy-key "$ABC123"; ## SECRET-DATA
                }
            }
        }
    }
    target-address md1 {
        address 198.51.100.240; ## Address of the mediation device receiving the traps
        port 162;
        tag-list mediation-8;
        target-parameters tp1;
    }
    target-parameters tpi {
        parameters {
            message-processing-model v3;
            security-model usm;
            security-level privacy;
            security-name mediation-device1; ## Name of the mediation device
        }
        notify-filter nf1;
```

```
    }
    notify n1 {
        type trap;
        tag mediation-8;
    }
    notify-filter nf1 {
        oid .jnxJsPacketMirrorMIB  include;
    }
}
view pkt-mirror-mib oid jnxJsPacketMirrorMIB include
```

Configure the radius-flow-tap service to support subscriber secure policy mirroring.

```
[edit services radius-flow-tap]
logical-system LS1 routing-instance RI1
snmp {
    notify-targets ip-address;
}
source-ipv4-address 198.51.100.255
```

RELATED DOCUMENTATION

- Subscriber Secure Policy Overview | 595
- Configuring SNMPv3 Traps for Subscriber Secure Policy Mirroring | 678
- Configure SNMPv3

PART 8

Configuring Stateless, Rule-Based Services Using Application-Aware Access Lists

- AACL Overview | **682**
- Configuring AACL Rules | **684**
- Example: Configuring AACL Rules | **690**
- Example: Configuring AACL Rule Sets | **692**
- Configuring Logging of AACL Flows | **693**

CHAPTER 40

AACL Overview

IN THIS CHAPTER

- AACL Overview | 682

AACL Overview

NOTE: Starting with Junos OS Release 12.1, all interface-style services are supported for dynamic Point-to-Point Protocol over Ethernet (PPPoE) subscribers on all MX Series routers with modular Modular Port Concentrators (MPCs).

The application-aware access list (AACL) service adds support for a new service that uses application names and groups as matching criteria for filtering traffic. AACL is a stateless, rules-based service that must be combined with application identification to enable policies to be applied to flows based on application and application group membership in addition to traditional packet matching rules. It is supported on MX Series routers equipped with Multiservices DPCs and on M120 or M320 routers equipped with Multiservices 400 PICs. Starting with Junos OS Release 11.3, AACL is supported on T320, T640, and T1600 routers also.

AACL is configured in a similar way to other rules-based services such as Network Address Translation (NAT), *class of service* (CoS), and stateful firewall. To configure AACL, include rule specifications for match criteria and actions at the `[edit services aacl]` hierarchy level. You can chain AACL rules along with other service rules by including them in a service-set definition at the `[edit services service-set]` hierarchy level, as previously documented.

There is one pair of related operational commands, `show/clear application-aware-access-list statistics`.

For more information on the operational command, see the CLI Explorer.

NOTE: Because the Junos OS extension-provider package framework lacks aggressive constraint checks, you should not set the `policy-db-size` statement at the `[edit chassis fpc` *`slot-number`* `pic` *`pic-number`* `adaptive-services service-package extension-provider]` hierarchy level

to a high value. For Junos Application Aware (previously known as dynamic application awareness) configurations, the recommended values for the extension-provider options at this hierarchy level are as follows:

- `control-cores = 1`
- `data-cores = 7`
- `object-cache-size = 1280` (for Multiservices 400 PIC and Multiservices DPC)
- `policy-db-size = 200`
- Include these `package` values: `jservices-aacl`

RELATED DOCUMENTATION

Configuring AACL Rules | **684**

Configuring AACL Rule Sets | **692**

Configuring Logging of AACL Flows | **693**

Example: Configuring AACL Rules | **690**

CHAPTER 41

Configuring AACL Rules

IN THIS CHAPTER

- Configuring AACL Rules | 684

Configuring AACL Rules

IN THIS SECTION

- Configuring Match Direction for AACL Rules | 685
- Configuring Match Conditions in AACL Rules | 686
- Configuring Actions in AACL Rules | 687
- Logging AACL Flows Based on Application | 688

To configure an AACL rule, include the `rule rule-name` statement at the `[edit services aacl]` hierarchy level:

```
rule rule-name {
    match-direction (input | output | input-output);
    term term-name {
        from {
            application-group-any;
            application-groups [ application-group-names ];
            applications [ application-names ];
            destination-address address <any-unicast>;
            destination-address-range low minimum-value high maximum-value;
            destination-prefix-list list-name;
            nested-applications [ nested-application-names ];
            nested-application-unknown
            source-address address <any-unicast>;
```

```
            source-address-range low minimum-value high maximum-value;
            source-prefix-list list-name;
        }
        then {
            (accept | discard);
            count (application | application-group | application-group-any | nested-application
| none);
            forwarding-class class-name;
            policer policer-name;
        }
    }
}
```

Each AACL 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 explain how to configure the components of AACL rules:

Configuring Match Direction for AACL 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 aacl 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 services PIC or DPC. When a packet is sent to the PIC or DPC, 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 services PIC or DPC. 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 on inside and outside interfaces, see *Configuring Service Sets to be Applied to Services Interfaces*.

On the PIC or DPC, 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 AACL Rules

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

```
from {
    application-group-any;
    application-groups [ application-group-names ];
    applications [ application-names ];
    destination-address address <any-unicast>;
    destination-address-range low minimum-value high maximum-value;
    destination-prefix-list list-name;
    nested-applications [ nested-application-names ];
    nested-application-unknown
    source-address address <any-unicast>;
    source-address-range low minimum-value high maximum-value;
    source-prefix-list list-name;
}
```

IPv4 and IPv6 source and destination addresses are supported. You can use either the source address or the destination address as a match condition, in the same way that you 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 the `source-prefix-list` statement in the AACL rule. For an example, see "Example: Configuring AACL Rules" on page 690.

If you omit the `from` term, the AACL rule 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 and application group definitions you have configured at the `[edit services application-identification]` hierarchy level; for more information, see the topics in "AACL Overview" on page 682.

- To apply one or more specific application protocol definitions, include the `applications` statement at the `[edit services aacl rule rule-name term term-name from]` hierarchy level.
- To apply one or more sets of application group definitions you have defined, include the `application-groups` statement at the `[edit services aacl 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 services application-identification]` hierarchy level; you cannot specify these properties as match conditions.

- To consider any application group defined in the database as a match, include the `application-group-any` statement at the `[edit services aacl rule rule-name term term-name from]` hierarchy level.
- To consider any nested application defined in the database a match, include the `nested-applications` statement at the `[edit services aacl rule rule-name term term-name from]` hierarchy level. Nested applications are protocols that run on a parent application. For example, if the Facebook application runs on the parent application junos:http, the nested application is junos:http:facebook.

Configuring Actions in AACL Rules

To configure AACL actions, include the `then` statement at the `[edit services aacl rule rule-name term term-name]` hierarchy level:

```
No Link Title {
    (accept | discard);
    (count (application | application-group | application-group-any | nested-application | none)
| forwarding-class class-name);
}
```

You must include one of the following actions:

- `accept`—The packet is accepted and sent on to its destination.
- `discard`—The packet is not accepted and is not processed further.

When you select `accept` as the action, you can optionally configure one or both of the following action modifiers. No action modifiers are allowed with the `discard` action.

- `count (application | application-group | application-group-any | nested-application | none)`—For all accepted packets that match the rules, record a packet count using AACL statistics practices. You can specify one of the following options; there is no default setting:

- application—Count the application that matched in the from clause.
- application-group—Count the application group that matched in the from clause.
- application-group-any—Count all application groups that match from application-group-any under the any group name.
- nested-application—Count all nested applications that matched in the from clause.
- none—Same as not specifying count as an action.

- forwarding-class *class-name*—Specify the packets' forwarding-class name.

You can optionally include a policer that has been specified at the [edit firewall] hierarchy level. Only the bit-rate and burst-size properties specified for the policer are applied in the AACL rule set. The only action application when a policer is configured is discard. For more information on policer definitions, see the Routing Policies, Firewall Filters, and Traffic Policers User Guide.

Logging AACL Flows Based on Application

You can now log AACL flows based on application. You can select a specific application or request information on unknown applications.

You can now configure AACL rules to match unknown applications. All existing actions that can apply to recognized applications can also apply to unknown applications. You can use the following statements at the [edit services aacl rule *rule-name* term *term-name* from] hierarchy level:

- application-group-any
- application-groups
- application-unknown
- applications
- nested-application-unknown
- nested-applications

The addition of matching application-unknown enables the specific logging of the input flows associated with applications that cannot be identified. Because logging is triggered by an input event, you must specify match-direction as input-output or input.

To configure logging of flows for AACL, include the match-direction input or match-direction input-output statement at the [edit services aacl rule *rule-name*] hierarchy level, include an applications or application-unknown statement at the [edit services aacl rule *rule-name* term *term-name* from] hierarchy level, and include

only one `log` statement at the `[edit services aacl rule rule-name term term-name then]` hierarchy level. The log statements can include any of the following options:

- `session-start`
- `session-end`
- `session-start-end-no-stats`
- `session-start-interim-end`
- `session-interim-end`
- `session-end`

RELATED DOCUMENTATION

AACL Overview | **682**

Configuring AACL Rule Sets | **692**

Configuring Logging of AACL Flows | **693**

Example: Configuring AACL Rules | **690**

CHAPTER 42

Example: Configuring AACL Rules

IN THIS CHAPTER

- Example: Configuring AACL Rules | 690

Example: Configuring AACL Rules

The following example shows an AACL configuration containing a rule with three terms using a variety of match conditions and actions:

```
[edit services aacl]
rule aacl-test {
    match-direction input;
    term term1 {
        from {
            source-address 10.0.1.1
            application test1;
        }
        then {
            accept;
        }
    }
    term term2 {
        from {
            source-address {
                any-unicast;
            }
            application test1;
        }
        then {
            discard;
        }
    }
```

```
    term term3 {
        from {
            source-address {
                any-unicast;
            }
            application test1 test2;
        }
        then {
            accept;
            count application;

        }
    }
}
```

RELATED DOCUMENTATION

AACL Overview | 682

Configuring AACL Rules | 684

CHAPTER 43

Example: Configuring AACL Rule Sets

IN THIS CHAPTER

- Configuring AACL Rule Sets | 692

Configuring AACL Rule Sets

The `rule-set` statement defines a collection of AACL 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 aacl]` 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.

RELATED DOCUMENTATION

- AACL Overview | 682
- Configuring AACL Rules | 684
- Configuring Logging of AACL Flows | 693
- Example: Configuring AACL Rules | 690

CHAPTER 44

Configuring Logging of AACL Flows

IN THIS CHAPTER

- Configuring Logging of AACL Flows | 693

Configuring Logging of AACL Flows

You can configure logging of AACL flows for a given application or for all unknown applications using AACL rules. You must set `match-direction` to `input` or `input-output` for logging to occur.

1. Create a rule and term.

```
user@host# edit services aacl rule rule-name term term-name
```

2. Specify selection of an application.

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

OR

Specify selection of all unknown applications.

```
[edit services aacl rule <variable>rule-name</variable > term <variable>term-name</variable>]
set from application-unknown
```

3. In the `then` statement, specify logging of input flow.

```
[edit services aacl rule rule-name term term-name]
user@host# set then log input-flows]
```

Example—Configuration of Logging of Input Flows for Unknown Applications

```
[edit services aacl rule aacl_rule5]
match-direction input-output;
    term t0 {
        from {
            application-unknown;
        }
        then {
            count application;
            log input-flow;
            accept;
        }
    }
```

Example—Setup of a Specific Log File

The following example shows how to direct the aacl flow log to a file other than the default syslog file on the Routing Engine file system.

```
[edit system syslog]
file aacl_log {
    external any;
    match aacl-flow-log;
}
```

RELATED DOCUMENTATION

- AACL Overview | **682**
- Configuring AACL Rules | **684**
- Configuring AACL Rule Sets | **692**
- Example: Configuring AACL Rules | **690**

PART 9

Remote Device and Service Management

- Configuring Remote Device Services Management | **696**
- Configuring TCP Port Forwarding for Remote Subscriber Services | **722**
- Configuring IPFIX Mediation for Remote Device Monitoring | **733**
- Collection and Export of Local Telemetry Data on the IPFIX Mediator | **743**

CHAPTER 45

Configuring Remote Device Services Management

IN THIS CHAPTER

- Remote Device Services Manager (RDSM) Overview | 696
- Configuring Remote Device Management for Service Provisioning | 715
- Reconfiguring a Remote Device for RDSM | 720
- Reloading a Dictionary File for RDSM | 721

Remote Device Services Manager (RDSM) Overview

IN THIS SECTION

- Remote Services | 698
- Process Flows for RDSM Provisioning and Deprovisioning | 699
- RDSM Dictionary for Implementing Service Actions | 704
- Additional Features for Use with an RDSM Access Model | 708
- Response to the External Authority by authd on Success or Failure | 709
- Operator Reconfiguration of Remote Devices | 712
- External Notification for Service Processing ERRMSG Events | 714
- Benefits of Remote Device Service Management | 715

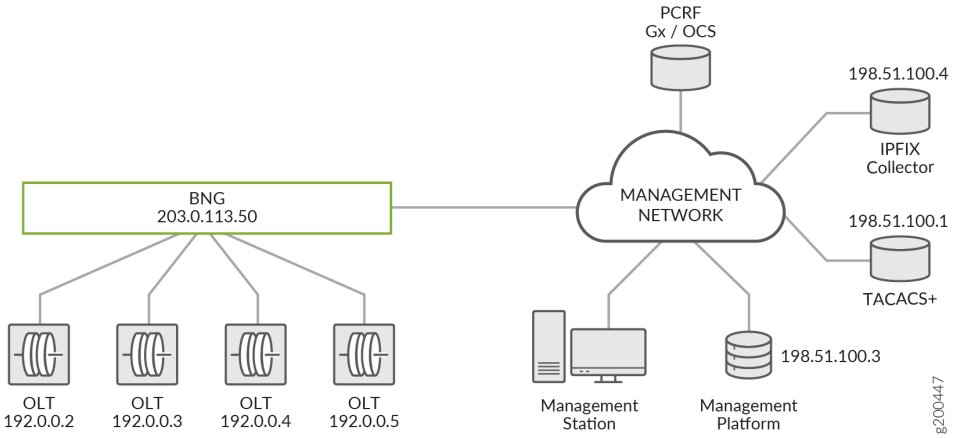
In some service provider use cases, subscriber services span both a broadband network gateway (BNG) and one or more access nodes. In order to minimize the number of network elements requiring operations support system/business support system (OSS/BSS) integration, the BNG and downstream access nodes are represented to back-office systems as a single, logical system. The service provider's back-office systems provide external configuration and management, authentication, and provisioning for subscriber services on the BNG and its downstream access nodes. To the back-office systems,

including PCRF and TACACS+ applications, the BNG and its nodes represent a single addressable network element. The BNG proxies for the downstream devices for service provisioning and deprovisioning.

Starting in Junos OS Release 18.3R1, MX Series routers used as a BNG support remote-device services by means of the remote device services manager (RDSM, using the rdmd daemon).

Figure 24 on page 697 shows a sample topology for an MX Series BNG using RDSM. The BNG is connected to OLTs that serve as the downstream, remote devices for provisioning subscriber services, in addition to their conventional role of terminating passive optical network (PON) access per individual subscriber access-lines. The OLTs are logical extensions to the BNG, so that the BNG and its downstream access nodes are presented to back-office systems as a single addressable network element. The BNG uses TCP port forwarding to mediate communications between the remote devices and the back-office system. For more information about TCP port forwarding for remote device management access, see "TCP Port Forwarding for Remote Device Management" on page 722.

Figure 24: Topology for Remote Device Management



The back-office management and provisioning system uses NETCONF XML protocol over SSH for tasks such as base configuration of the remote device before subscriber negotiation begins, configuration of Layer 2 data paths for new subscribers, displaying remote device status, and troubleshooting the remote device. The BNG demultiplexes requests from the management system to the remote devices. Multiple NETCONF sessions can exist to a single remote device.

In this sample topology, the system includes a management platform, PCRF, TACACS+ server, and an IPFIX collector:

- The PCRF sources the subscriber services that are provisioned locally on the MX BNG locally and remotely on the OLTs.

- The TACACS+ server is used to authenticate and validate access to the remote device, perform system accounting, and control operator access. The remote device dynamically initiates a TACACs+ TCP session in response to NETCONF protocol configuration from an external management platform or station. The BNG multiplexes requests from the remote devices to the TACACS+ server.

  For remote device access from the back-office system, the server initiates TACACS+ authentication for the following conditions:

  - The BNG initiates service configuration for a remote device. The TACACS+ server authenticates the session when the NETCONF TCP socket used by the BNG to provision or deprovision the remote service is opened. After authentication, the session is maintained without authentication or authorization for each remote procedure call (RPC) used for the service action.

  - The external management station is used to configure the remote device or access it for monitoring (`show` commands) or troubleshooting.

- The IPFIX collector receives records containing system and connection-level statistics and other information from the MX BNG, which operates as an IPFIX mediator between the OLTs (IPFIX exporters) and the external IPFIX collector. The BNG proxies for the downstream devices. It acts as an IPFIX collector to receive data from the remote devices and as an IPFIX exporter to send data upstream over a single TCP or TLS session to the collector. For more information about using the BNG as an IPFIX mediator, see "IPFIX Mediation on the BNG" on page 733.

Remote Services

The MX BNG represents a single point of management to external authority for all subscriber services, local and remote. The remote services are also represented by locally configured dynamic service profiles that are referenced by external authority in the same way as local services on the BNG. Consequently, there is a consistent interface between external authority and the BNG for all service actions. The NETCONF XML Management Protocol is used for provisioning and deprovisioning the remote services.

Local subscriber services are defined by dynamic service profiles with zero or more arguments to satisfy subscriber-specific policies. External authorities, such as PCRF, generally use a referential model to provide services. The PCRF charging rule specifies the name of the dynamic service profile and argument values that are applied during subscriber negotiation for service provisioning (activation) or as an update after the subscriber is active. The service is presented to the remote device by the RDSM XML dictionary for that device to parse, interpret, and apply, allowing the charging-rule or service from external authority to be opaquely passed to the remote device with minimal processing. The remote service profile might include one or more variables to define service parameters.

However, remote services can also be applied in a non-referential manner. In this case, the external authority specifies the remote service referentially as it would for a local service. The remote service profile includes one or more variables to define service parameters. The RDSM then uses the data dictionary assigned to the remote device to configure the service on that device. The content of the

RDSM dictionary for a device is different depending on whether the service provider uses the referential or non-referential method.

The remote dynamic service profile is very lightweight compared to a local service profile, which can include a large number of configuration stanzas. A remote dynamic service profile contains only two things:

- You must specify that the dynamic profile type is `remote-device-service`. That configuration prevents the profile from being used as a local service profile. This means that you cannot configure a dynamic service profile to be dual-purpose (both local and remote).
- The remote service profile can optionally include a variable stanza to pass argument values to the remote device. The variable stanza can be used for either the referential or non-referential methods.

Any additional configuration fails commit check. Because the remote service profile is so specific, a dedicated service profile is required for each remote service. For the external authority, this means that each remote service requires a separate PCRF charging rule.

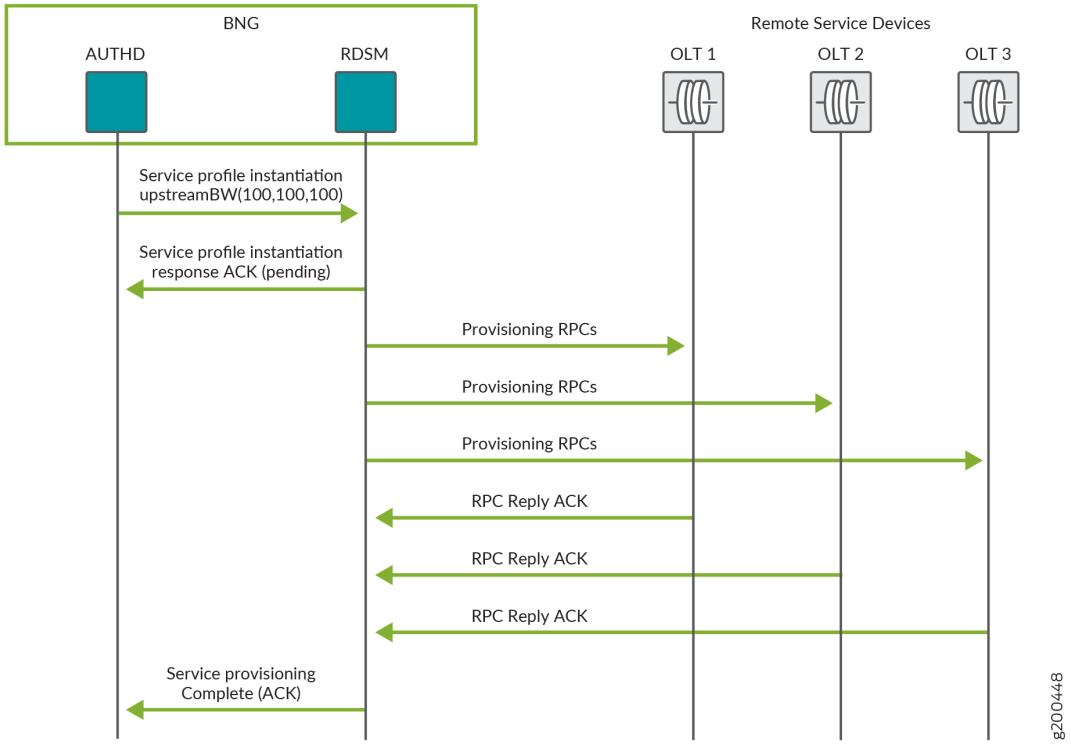
Process Flows for RDSM Provisioning and Deprovisioning

Subscriber services are provisioned and deprovisioned as follows:

- Provisioned during subscriber login. The services can be sourced from the PCRF in response to initiation with Gx CCR-I/CCR-A message exchanges.
- Deprovisioned during subscriber logout.
- Provisioned or deprovisioned for active subscriber sessions in response to external authority, such as Gx RAR messages from the PCRF.

Figure 25 on page 700 shows the process flow when RDSM successfully provisions services on three eligible remote devices, OLT1, OLT2, and OLT3, by instantiating the upstreamBW service profile.

Figure 25: RDSM Service Provisioning on a Remote Device: Successful Subscriber Negotiation Flow



1. Service provisioning begins when a subscriber logs in and authd sends a request to RDSM to instantiate the remote service profile on eligible remote devices during the negotiation.
2. RDSM establishes a list of remote devices that are eligible for the service to be provisioned:
   - The Layer 2 access domain for the device must match the subscriber location. The access domain consists of a configured list of VLAN ranges or individual VLAN IDs. The subscriber's outer VLAN tag must be on this list.
   - The NETCONF TCP connection to the device must be up. Although a device in the down state is not eligible for provisioning, it might be available for reconfiguration if it transitions later to the up state.
3. RDSM performs an initial validation before it responds to the remote service profile instantiation request:
   - When validation passes, RDSM sends a service profile instantiation pending ACK response to authd. The service provisioning is now pending.
   - If validation fails, RDSM returns a NACK response to authd and abandons service provisioning.

The validation checks performed by RDSM typically do not fail for active subscriber sessions. Reasons for failure include the following:

- No remote device has a subscriber location that matches the access domain.
- The dictionary located on the BNG does not include an entry for the requested remote service profile. Consequently there are no RPCs to provide the service variables and install the service.

4. RDSM resolves any required parameters for each remote device; at a minimum, this includes the subscriber identifier.
5. RDSM then uses the dedicated NETCONF session to each of the eligible devices to issue a series of RPC calls as specified in the dictionary for provisioning the service.

   Service provisioning takes place in parallel for the eligible devices. Provisioning fails for a device when either of the following occurs:

   - The RDSM receives an explicit error for any RPC call.
   - The response times out.

   The following ERRMSG event is logged in either case:

   `remote device` *`device-name ip-address`* `service` *`service-name`* `provisioning failed for subscriber` *`subscriber-id`*

6. Remote devices that are successfully provisioned return an ACK response to RDSM.

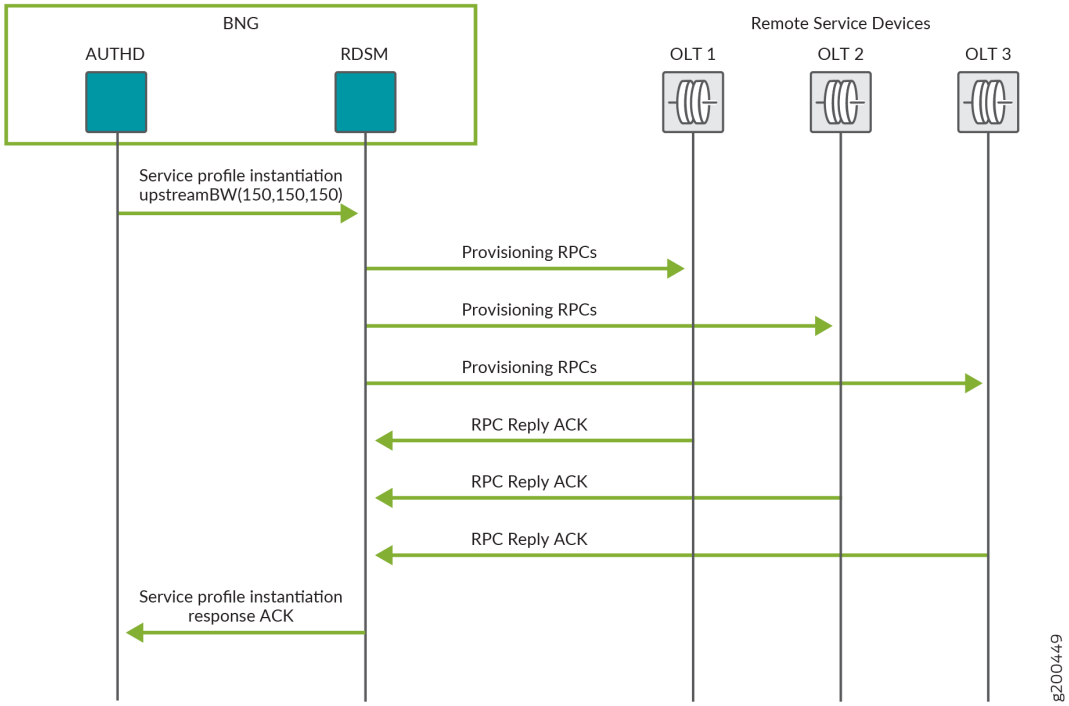
   If one or more remote devices fails to be provisioned, RDSM rolls back the service on every remote device that was successfully provisioned. RDSM uses the dedicated NETCONF session to each of these devices to issue a series of RPC calls as specified in the dictionary for deprovisioning the service.

7. RDSM sends an out-of-band notification to authd to report whether the remote service was provisioned on the remote devices.

   - When provisioning is successful for all remote devices, RDSM sends a service provisioning complete response to authd.
   - If one or more of the eligible remote devices fails to be provisioned, RDSM reports a provisioning failure to authd.

Figure 26 on page 702 shows the process flow when RDSM successfully updates subscriber services on three eligible remote devices, OLT1, OLT2, and OLT3 by instantiating the upstreamBW service profile with different parameter values than were used during login.

Figure 26: RDSM Service Provisioning on a Remote Device: Subscriber Update Flow



Updating subscriber services begins when authd sends a request to RDSM to instantiate the remote service profile to update the service. The process flow is the same as for the subscriber login flow, except that RDSM does not respond to the instantiation request until all processing required to provision the service is complete. That means that when the validation check passes, RDSM does not send a service profile instantiation pending ACK response to authd; if validation fails, RDSM does return a NACK response to authd and abandons service deprovisioning.

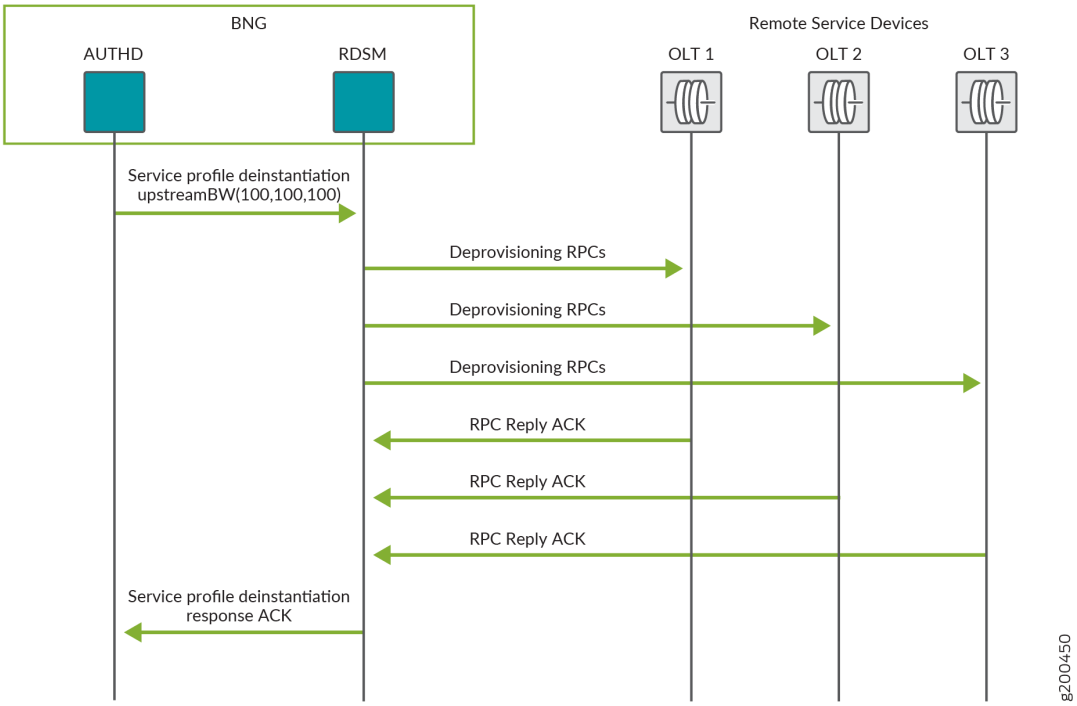
Figure 27 on page 703 shows the process flow when RDSM successfully deprovisions services to update the three eligible remote devices, OLT1, OLT2, and OLT3, by deinstantiating the upstreamBW service profile.

The deprovisioning process flow is the same for both a subscriber logout and an update request from authd.

RDSM does not respond to authd until all required processing to deprovision the service has completed (including any retry of failures); this allows subscriber logout to proceed regardless of the deprovisioning outcome.

Service deprovisioning typically does not fail; if it does, then you may have to take some corrective action on the remote device for deprovisioning to succeed.

Figure 27: RDSM Service Deprovisioning on a Remote Device: Subscriber Logout and Update Flow



1. Service deprovisioning begins when either of the following occurs:
   - A subscriber logs out and authd sends a request to RDSM to deinstantiate the remote service profile on eligible remote devices.
   - authd sends an update request to RDSM to deinstantiate the remote service profile on eligible remote devices.
2. RDSM maintains a list of remote devices that are provisioned with the service. If the NETCONF TCP connection to the device is down, deprovisioning is not attempted because it is assumed to have occurred by some other means. For example, the device may have been reconfigured with a default, baseline configuration and subsequent operator action initiated reconfiguration by the BNG for all active subscriber services.
3. RDSM performs an initial validation before it responds to the remote service profile deinstantiation request:
   - When validation passes, RDSM does not send a response to authd.
   - If validation fails, RDSM returns a NACK response to authd and abandons service deprovisioning. Reasons for a validation failure include the following:
     - No configured remote device is in the up state.

- The dictionary located on the BNG does not include an entry for the requested remote service profile or deprovisioning action. Consequently there are no RPCs to provide the service variables and remove the service.

The validation checks performed by RDSM typically do not fail for active subscriber sessions.

4. RDSM resolves any required parameters for each remote device; at a minimum, this includes the subscriber identifier.

5. RDSM then uses the dedicated NETCONF session to each of the eligible devices to issue a series of RPC calls as specified in the dictionary for deprovisioning the service. Service deprovisioning takes place in parallel for the eligible devices. Deprovisioning fails for a device when either of the following occurs:

   - The RDSM receives an explicit error for any RPC call.
   - The response times out.

   In either case, RDSM retries the deprovisioning action up to 5 times, at 5-second intervals. If the attempts all fail, then the following ERRMSG event is logged in either case:

   `remote device` *`device-name ip-address`* `service` *`service-name`* `de-provisioning failed for subscriber` *`subscriber-id`*

6. Remote devices that are successfully deprovisioned return an ACK response to RDSM.

7. RDSM sends an out-of-band notification to authd to report whether the remote service was deprovisioned on the remote devices.

   - When deprovisioning is successful, RDSM sends a service deprovisioning complete response to authd, which then completes the subscriber logout.

     In the case of an update request rather than a subscriber logout, RDSM sends a service profile deinstantiation complete response to authd, which completes the service session clean-up.

   - If one or more of the eligible remote devices fails to be deprovisioned, RDSM reports a deprovisioning failure to authd.

RDSM Dictionary for Implementing Service Actions

An XML dictionary stored locally on the BNG is an integral component of remote device service management. Each remote service provisioned by the external authority must have an entry in an RDSM dictionary on the BNG. The dictionary translates the PCRF-sourced charging rule to a set of vendor-specific remote procedure calls (RPCs) in the entry associated with the service. The RPCs then provision or deprovision the service. Because the RPCs are vendor-specific, so is the dictionary. This means that separate dictionaries are required for each vendor's remote device. For a given vendor's devices, different software releases on the devices may require different dictionaries as well.

The dictionary format is sufficiently flexible to support both referential services and non-referential services, where:

- A referential service means that the entire service, including arguments, is presented opaquely to the remote device as received from external authority via the RDSM dictionary. The dynamic service profile can include a variable stanza that is used by the dictionary during translation of the arguments. The remote device parses, interprets and applies the arguments on its own without any interpretation or parsing by the BNG.
- A non-referential service means that all arguments supplied by the external authority must be resolved and provided to the remote device individually by one or more RPCs. In this case, the dynamic service profile may require a variable stanza that is used by the dictionary during translation of the arguments.

In either case, the dictionary must specify the means—typically a Layer 2 location—to identify the subscriber suitable for the remote device to distinguish one subscriber from another.

The XML RDSM dictionary has the following general format:

```
<junos-rdm-dictionary>
  <junos-rdm-parameters>
      <junos-rdm-parameter>
          <junos-rdm-name>...</junos-rdm-name>
          <junos-rdm-source>...</junos-rdm-source>
          <junos-rdm-index>...</junos-rdm-index>
      </junos-rdm-parameter>

      <junos-rdm-parameter>
...
      </junos-rdm-parameter>
  </junos-rdm-parameters>

  <junos-rdm-services>      <junos-rdm-service>
          <junos-rdm-name>...</junos-rdm-name>
          <junos-rdm-provision>
              <junos-rdm-service-configuration>
...
              </junos-rdm-service-configuration>
          </junos-rdm-provision>
          <junos-rdm-deprovision>
              <junos-rdm-service-configuration>
...
              </junos-rdm-service-configuration>
          </junos-rdm-deprovision>
```

```
        </junos-rdm-service>
    </junos-rdm-services>

    <junos-rdm-open-configuration>
        <junos-rdm-rpc>...</junos-rdm-rpc>
        <junos-rdm-rpc>...</junos-rdm-rpc>
    </junos-rdm-open-configuration>

    <junos-rdm-edit-configuration>
        <junos-rdm-rpc>...</junos-rdm-rpc>
        <junos-rdm-rpc>...</junos-rdm-rpc>
        <junos-rdm-rpc>...</junos-rdm-rpc>
    </junos-rdm-edit-configuration>

    <junos-rdm-commit-configuration>
        <junos-rdm-rpc>...</junos-rdm-rpc>
        <junos-rdm-rpc>...</junos-rdm-rpc>
    </junos-rdm-commit-configuration>

    <junos-rdm-close-configuration>
        <junos-rdm-rpc>...</junos-rdm-rpc>
        <junos-rdm-rpc>...</junos-rdm-rpc>
    </junos-rdm-close-configuration>
</junos-rdm-dictionary>
```

Table 49 on page 706 defines the individual components of the dictionary.

Table 49: Definitions of XML Dictionary Components

junos-rdm-parameters	Parameter block that lists individual parameters that configure the service.
junos-rdm-parameter	Individual parameter.
junos-rdm-name	In the parameter block, this element identifies the subscriber on the remote device or the PCRF argument. Use the subscription-id for the subscriber and the name of the argument for any argument specified in the PCRF.

Table 49: Definitions of XML Dictionary Components *(Continued)*

junos-rdm-source	Source of the parameter value: • subscriber-session when the value is sourced from the SDB session. • service-profile when the value is sourced from the service profile argument.
junos-rdm-index	Index, such as an enumerated type value, that resolves the parameter from the specified source. The subscriber-session source requires this to map the parameter to an SDB attribute used to resolve the parameter value. For example, for some use cases, the PCRF subscription-id is stored in the subscriber SDB entry that is referenced by an index (attribute type) to resolve this parameter.
junos-rdm-services	Service block that lists one or more remote services supported by the device.
junos-rdm-service	Individual remote service defined by service name, provisioning configuration, and deprovisioning configuration.
junos-rdm-name	In the service block, this element is the name of the service. It is the base service name, without arguments, of the service sourced from the PCRF.
junos-rdm-provision	Provisioning block that includes provisioning configuration.
junos-rdm-deprovision	Deprovisioning block that includes deprovisioning configuration.

Table 49: Definitions of XML Dictionary Components *(Continued)*

junos-rdm-service-configuration	Service configuration that includes one or more RPCs to provision or deprovision the service. When arguments are specified in the PCRF service for provisioning, the RPCs include those arguments.
junos-rdm-open-configuration	Block that includes zero or more RPCs to begin configuration of the remote device.
junos-rdm-edit-configuration	Block that includes one or more RPCs to edit the configuration and apply service provisioning or deprovisioning actions to the device in bulk, by referencing the junos-rdm-provision or junos-rdm-deprovision block for the specified service. The configuration for each service that is part of the bulk update to the remote device is included.
junos-rdm-commit-configuration	Block that includes zero or more RPCs to commit the edits to the remote device.
junos-rdm-close-configuration	Block that includes zero or more RPCs to end configuration of the remote device.
junos-rdm-rpc	Individual RPC to configure the remote device.

For remote device configuration, the edit configuration is always required to provision or deprovision the service. In some use cases, the open, commit, and close configuration blocks might be optional.

Additional Features for Use with an RDSM Access Model

The features in this section are not required for RDSM, but may be useful in certain use cases or topologies.

A locally generated username is used in interactions with an external authority to authenticate dynamic VLAN, DHCPv4, and DHCPv6 subscribers. Typically, subscriber VLAN tags are included in the username by configuring the `interface-name` option for the `username-include` statement.

Similarly, subscriber VLAN tags are included in the subscription identifier for PCRF interactions by configuring the `interface-name` option for the `subscription-id-data-include` statement.

By convention, the interface name has the following format in both cases:

```
underlying-IFD-name:outer-vlan-tag[-inner-vlan-tag]
```

For some use cases with the RDSM access model, the outer VLAN tag is unique across the system. This means that you can use a different format that excludes the underlying IFD name:

```
outer-vlan-tag[-inner-vlan-tag]
```

To generate the username format without the underlying IFD name, you specify the `vlan-tags` option instead of the `interface-name` option with the `username-include` statement. See *Configuring VLAN Interface Username Information for AAA Authentication* and *Creating Unique Usernames for DHCP Clients* for more information.

To generate the subscription ID format without the underlying IFD name, you specify the `vlan-tags` option instead of the `interface-name` option with the `subscription-id-data-include` statement. See *Configuring the PCRF Partition* for more information.

Some customer networks might have more than one deployment model or use case that results in the MX Series BNG for each case interacting with the same PCRF back-end. In this situation, you might need to distinguish between the use cases for the PCRF.

The Diameter Capability Exchange messages between peers carry the Diameter Product-Name AVP. You can configure nondefault values for the use cases so the PCRF can discriminate between the messages. See *Messages Used by Diameter Applications* and *Diameter AVPs and Diameter Applications* for more information.

Response to the External Authority by authd on Success or Failure

How authd responds to the external authority depends on the following:

- The operation being performed for example, provisioning during subscriber login versus updating an existing subscriber session.
- The external authority, Gx (PCRF).

Table 50 on page 710 describes how the authd response varies when the service provisioning or deprovisioning actions are successful.

Table 50: How authd Responds to External Authority When Service Actions Succeed

Operation	Gx
Login	authd initiates CCR-I/CCA-I message exchange to provision the subscriber session. When all services in the CCA-I are provisioned, authd sends a CCR-U message that indicates the service is active for each charging-rule in the CCA-I. Status reporting for local dynamic services is delayed until remote services provisioning completes.
Update	Deprovisioning is applied before provisioning for services included in the same PCRF RAR message. When deprovisioning and provisioning is completed for all service actions included in the PCRF RAA, authd sends an RAA response with a Rule-Report that indicates the service inactive/active state for each charging rule specified in the RAR. Status reporting for local dynamic services is delayed until remote services processing completes.
Logout	authd initiates a CCR-T/CCA-T message exchange to notify PCRF of subscriber termination. authd initiates deprovisioning for all services configured for the subscriber session.
Service device in the up state after the `reconfigure` command is issued	authd takes no further action when it receives an out-of-band notification from RDSM that the service action succeeded. For example, it does not send a CCR-U message that indicates the service is active for the corresponding charging rule.

Table 51 on page 711 describes how the authd response varies when the service provisioning or deprovisioning actions fail.

Table 51: How authd Responds to External Authority When Service Actions Fail

Operation	Gx
Login	authd initiates a CCR-I/CCA-I exchange to provision the subscriber session. When authd receives notification of failure in the service profile instantiation response or out-of-band from RDSM, authd stops processing any remaining services. authd sends a CCR-U message that reports the following: • Service is active for each charging-rule in the CCA-I that successfully provisioned • Service is inactive for the charging rule in the CCA-I that failed provisioning. • Service is inactive for all charging rules not processed because of the failure. authd allows the subscriber session negotiation to complete and reach the active state.
Update	Deprovisioning is applied before provisioning for services included in the same PCRF RAR message. The process varies depending on the actions that fail. • When authd receives notification of failure in the service profile deinstantiation response, meaning that RDSM has performed all retries without success, authd continues to process the next service action. This means that when only service deprovisioning fails, the update proceeds and completes. • When authd receives notification of failure in the service profile instantiation response, authd stops processing any remaining services. All provisioned and deprovisioned services in the request are rolled-back. That means that services that were successfully provisioned are now deprovisioned. Services that were successfully deprovisioned are now reprovisioned. When all rollback actions are completed, authd sends an RAA response with a Rule-Report that indicates the service inactive/active state for each charging rule specified in the RAR. This means that reprovisioned charging-rules are reported as active and deprovisioned charging-rules are reported as inactive.

Table 51: How authd Responds to External Authority When Service Actions Fail *(Continued)*

Operation	Gx
Logout	authd initiates a CCR-T/CCA-T message exchange to notify PCRF of subscriber termination. authd initiates deprovisioning for all services configured for the subscriber session. When authd receives notification of failure in the service profile instantiation response or out-of-band from RDSM, authd continues with the logout, including deprovisioning any remaining services.
Last service device in down state after the `reconfigure` command is issued	authd takes no further action when it receives an out-of-band notification from RDSM that the service action failed. For example, it does not send a CCR-U that indicates the service is inactive for the corresponding charging rule. Affected subscriber sessions are maintained.

Operator Reconfiguration of Remote Devices

In some circumstances, you might need to manually provision services on a remote device to resynchronize the device with all matching subscriber services that are active and configured on at least one other remote device. Manual provisioning is required in the following scenarios:

- A new remote device is connected to the BNG after one or more subscriber sessions have been negotiated on other remote devices and remote services have been provisioned on those devices.
- The NETCONF session to a remote device with one or more provisioned remote services transitions to the down state, then later recovers and transitions back to the up state. This is effectively the same as a new device being connected to the BNG.

After the NETCONF session is established to the remote device in either of these situations, an ERRMSG event is logged that the device is up. No remote services are currently provisioned on the device. RDSM establishes a list of subscriber remote services that are eligible to be provisioned on the device. These services must be either active or in the process of being provisioned. A separate ERRMSG event is logged, indicating that services are pending reconfiguration:

`remote device` *`device-name ip-address`* `has` *`number`* `services pending reconfiguration`

You use the `request services remote-device-management reconfigure service-device` command to provision all active (or in process) subscriber services that map to the access domain associated with the device. The reconfiguration request triggers bulk provisioning of services on the device. If the provisioning of one

service fails, the entire bulk provisioning is considered a failure and any successfully provisioned services are rolled back. In this case you have to issue the command again. The rollback applies only to each bulk provisioning attempt, so you can control the effects of a bulk provisioning failure by setting a bulk limit.

NOTE: The remote device is eligible to be automatically provisioned with subscriber services without operator intervention for subscriber logins that occur after the NETCONF session is established.

You can issue a reconfiguration request at any time when the remote device is up. When remote device reconfiguration begins, any new service actions resulting from new subscriber negotiation or existing subscriber update or logout are delayed for the remote device until reconfiguration completes. Also, a reconfiguration request may be performed at any time when the remote device is up. This means that a remote device may be connected to the network and accept new subscriber services provisioning before existing subscribers are provisioned by the reconfiguration request.

The following steps show the RDSM process flow for reconfiguration requests:

1. RDSM maintains a list of remote devices that are provisioned with the service. If the NETCONF TCP connection to the device is down, deprovisioning is not attempted because it is assumed to have occurred by some other means. For example, the device may have been reconfigured with a default, baseline configuration and subsequent operator action initiated reconfiguration by the BNG for all active subscriber services.

2. RDSM performs the following as a bulk operation, where the bulk size maybe up to total number of subscriber services to be provisioned:

   a. Validates the service before it responds to the reconfiguration request. For example, validation fails when the dictionary located on the BNG does not include an entry for the requested remote service profile or provisioning action, because there are no RPCs to provide the service variables and add the service.

   b. Resolves any required parameters for each remote device; at a minimum, this includes the subscriber identifier.

   c. Uses the dedicated NETCONF session to the remote device to issue a series of RPC calls as specified in the dictionary for provisioning the service. Provisioning fails for a device when either of the following occurs:

      - The RDSM receives an explicit error for any RPC call.
      - The response times out.

      In either case, RDSM rolls back all service that were successfully provisioned by the bulk operation, reconfiguration is abandoned and RDSM logs the following ERRMSG event:

      ```
      remote device device-name ip-address reconfiguration failed
      ```

3. If provisioning completes for all the subscriber services on the remote device, RDSM logs the following ERRMSG event:

```
remote device device-name ip-address reconfiguration succeeded
```

External Notification for Service Processing ERRMSG Events

Table 52 on page 714 lists the ERRMSG events that authd can communicate to external management systems and the information that is included in the notifications. Successful remote service actions are only reported to an external authority and do not generate an ERRMSG log.

Table 52: Information Included in External Notifications for ERRMSG Events

ERRMSG Event	Device Name	IP Address	Current State	Number of Services Pending Reconfiguration	Service Name	Subscriber Identifier
Remote device status change from up to down or down to up	✓	✓	✓	–	–	–
Remote device has services pending reconfiguration	✓	✓	–	✓	–	–
Remote device reconfiguration completion (success or failure)	✓	✓	✓	–	–	–
Subscriber remote service provisioning failure	✓	✓	–	–	✓	✓
Subscriber remote service deprovisioning failure	✓	✓	–	–	✓	✓

Benefits of Remote Device Service Management

- Enables topologies where subscriber services span both the MX Series BNG and its access nodes to form a single, logical system.
- Simplifies BNG and remote device configuration and management in topologies that use external management and provisioning systems. The remote devices typically have private addresses unknown to the external system, so the external system addresses only the MX Series BNG.
- Adds a new service profile type for remote services to easily differentiate remote and local services.

RELATED DOCUMENTATION

Configuring Remote Device Management for Service Provisioning | 715

Reconfiguring a Remote Device for RDSM | 720

Reloading a Dictionary File for RDSM | 721

Configuring Remote Device Management for Service Provisioning

You must also configure the back-office system that provides the external authority and management platform for remote device service management. That configuration is outside the scope of this topic. Consult the vendor documents for your back-office equipment.

You must configure both dynamic service profiles and remote devices. A dynamic service profile is identified for RDSM by configuring the profile type as remote-device-service. This profile type prevents the profile from being applied locally on the router. It is limited to application on an external device by RDSM. The external authority, such as PCRF can reference this profile to provision or deprovision services on the remote device.

The remote device configuration includes the device IP address and the dictionary path. The remote device must have an entry in an XML dictionary hosted on the MX BNG. The dictionary translates the service action instructions from the external authority to a set of vendor-specific remote procedure calls (RPCs) in the entry associated with the service. The RPCs then provision or deprovision the service.

Finally, you can configure several parameters for the provisioning method, the NETCONF XML protocol. You must configure the username and password used to access the remote device. Other parameters are optional.

NOTE: Although the following procedure shows only configuration at the `[edit system services]` hierarchy level, and therefore the default routing instance, you can also configure RDSM at the `[[edit routing-instances routing-instance-name system services]` hierarchy level.

To configure remote device service management:

1. Configure one or more dynamic service profiles. Specify that the dynamic service profile containing this statement is not applied locally to the router. Instead, it is applied to an external device by means of the remote device services manager daemon (rdmd). It enables an external authority, PCRF to reference the dynamic service profile to provision or deprovision services (charging rules) on the remote device.

```
[edit dynamic-profiles profile-name]
user@host# set profile-type remote-device-service
```

2. Configure one or more devices for remote services.

```
[edit system services remote-device-management]
user@host# edit service-device device-name
```

3. (Optional) Configure the Layer 2 access domain for the remote device.

```
[edit system services remote-device-management service-device device-name]
user@host# set access-domain vlan-id-list [vlan-id-low-vlan-id-high vlan-id
```

4. Configure the address for the remote device.

```
[edit system services remote-device-management service-device device-name]
user@host# set address ip-address
```

5. Specify the absolute file path for the XML dictionary.

```
[edit system services remote-device-management service-device device-name]
user@host# set dictionary absolute file path
```

6. Specify the provisioning method (only NETCONF XML protocol is supported).

```
[edit system services remote-device-management service-device device-name]
user@host# edit provisioning-method netconf
```

7. Configure the provisioning parameters for the NETCONF protocol. You must specify the username and password; all other options have default values.

   a. Specify the name used to access the remote device during service management. The maximum length of the name is 64 bytes.

   NOTE: If you change the username when any active subscriber services are mapped to the device, the change takes effect only when the device reconnects.

```
[edit system services remote-device-management service-device device-name provisioning-
method netconf]
user@host# set user-name name
```

   b. Specify the password used by the NETCONF protocol to access the remote device during service management. The maximum length of the password is 64 bytes.

   NOTE: If you change the password when any active subscriber services are mapped to the device, the change takes effect only when the device reconnects.

```
[edit system services remote-device-management service-device device-name provisioning-
method netconf]
user@host# set password password
```

   c. (Optional) Specify the period during which multiple services are provisioned or deprovisioned based on the assigned dictionary before the configuration is committed to the service device. When the interval times out, the service actions are committed in bulk before additional actions for the device can take place.

NOTE: You can use the `bulk-interval` and `bulk-limit` options together to optimize your service device configuration during scaled subscriber negotiation and service provisioning or subscriber termination and service deprovisioning.

```
[edit system services remote-device-management service-device device-name provisioning-
method netconf]
user@host# set bulk-interval milliseconds
```

d. (Optional) Specify how many services can be provisioned or deprovisioned during the bulk interval before the configuration is committed to the device.

```
[edit system services remote-device-management service-device device-name provisioning-
method netconf]
user@host# set bulk-limit number
```

e. (Optional) Specify how long RDSM waits between successive attempts to establish a NETCONF session with the remote device.

```
[edit system services remote-device-management service-device device-name provisioning-
method netconf]
user@host# set connection-retry-interval seconds
```

f. (Optional) Specify the TCP port number for the NETCONF session with the remote device.

```
[edit system services remote-device-management service-device device-name provisioning-
method netconf]
user@host# set port port-number
```

g. (Optional) Specify how many services can be provisioned or deprovisioned as a result of a reconfiguration before the configuration is committed to the service device. When the limit is reached, the service actions are committed in bulk before additional actions for the device can take place.

```
[edit system services remote-device-management service-device device-name provisioning-
method netconf]
user@host# set reconfigure-bulk-limit number
```

h. (Optional) Specify the period during which the device must respond to an attempt to provision or deprovision a service. The timeout is a failure equivalent to an explicit failure response received from the device.

```
[edit system services remote-device-management service-device device-name provisioning-
method netconf]
user@host# set response-timeout seconds
```

i. (Optional) Specify how many consecutive response timeouts can occur before the BNG takes action. The default action is to close and reopen the NETCONF connection.

```
[edit system services remote-device-management service-device device-name provisioning-
method netconf]
user@host# set response-timeout-count number
```

Table 53 on page 719 lists commands you can use to view information about your RDSM configuration and operation.

Table 53: show Commands for Remote Device Services Management

Command	Description
show remote-device-management service-devices	Display information about all remote service devices or a specific remote service device.
show remote-device-management services	Display information about all service sessions or a specific service session on remote service devices.
show remote-device-management statistics	Display a global summary of service statistics for all remote devices or detailed statistics for a specific remote service device.
show remote-device-management subscribers	Display information about service sessions for all subscriber sessions or about all service sessions for a specific subscriber session on remote service devices.
show remote-device-management summary	Display summary information about the remote service devices, such as session state and service state.

You can use the *clear remote-device-management statistics* command to clear service statistics for all remote devices globally or statistics for a specific remote service device.

RELATED DOCUMENTATION

Remote Device Services Manager (RDSM) Overview | 696

Reconfiguring a Remote Device for RDSM | 720

Reloading a Dictionary File for RDSM | 721

Reconfiguring a Remote Device for RDSM

In some circumstances you might need to reconfigure a remote device to manually provision all active subscriber services matching the access domain (list of VLAN ranges and IDs) to which this remote device belongs. The reconfiguration resynchronizes the device with all active (or in process) subscriber services that map to the access domain associated with the device.

For example, if a new remote device is connected to the BNG after subscriber sessions have been brought up on other remote devices in the same access domain and remote services have been provisioned on the devices. The new device is not provisioned at this point, and you would like it be provisioned as if it had been connected during the original service provisioning.

Another situation occurs when the NETCONF session to a provisioned remote device transitions to the down state and then back to the up state. From the perspective of the BNG, this is the same as if the device is new and connected to the BNG for the first time.

You can issue a reconfiguration request at any time when the remote device is up. Reconfiguration provisioning of services occurs in bulk. If the provisioning of one service fails, the entire bulk provisioning is considered a failure and any successfully provisioned services are rolled back. You must issue the command again.

To reconfigure service provisioning for a device:

- Specify the device to be reconfigured.

```
user@host> request services remote-device-management reconfigure service-device device-name
```

The command indicates whether the action succeeds or fails.

RELATED DOCUMENTATION

Configuring Remote Device Management for Service Provisioning | 715

Remote Device Services Manager (RDSM) Overview | 696

Reloading a Dictionary File for RDSM

You can reload the vendor-specific dictionary to the RDSM database on the BNG by specifying the absolute file path. An example absolute path is **/var/home/dict/remote-device.xml**. The path must end with the .xml extension and not exceed 127 characters.

The dictionary defines the set of NETCONF XML protocol commands required to provision, deprovision, and roll back a subscriber service for a remote device. The reload affects all remote service devices that are configured with this dictionary. When you modify an existing dictionary, this is how you apply the updated file.

To reload a dictionary:

- Specify the path for the dictionary to be reloaded.

```
user@host> request services remote-device-management reload-dictionary absolute file path
```

The command indicates whether the action succeeds or fails. A typical cause for failure is when there is an active remote device configured with that dictionary and the device has an active subscriber service.

RELATED DOCUMENTATION

Remote Device Services Manager (RDSM) Overview | 696

Configuring Remote Device Management for Service Provisioning | 715

CHAPTER 46

Configuring TCP Port Forwarding for Remote Subscriber Services

IN THIS CHAPTER

- TCP Port Forwarding for Remote Device Management | 722
- Configure TCP Port Forwarding for Remote Device Management | 725
- Tracing TCP Port Forwarding Events for Troubleshooting | 729

TCP Port Forwarding for Remote Device Management

IN THIS SECTION

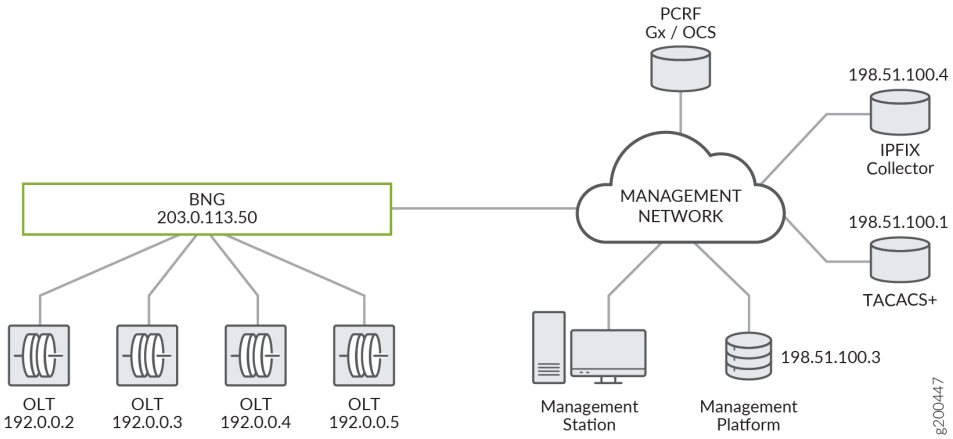
- Benefits of TCP Port Forwarding | 725

Port forwarding is a method that enables a router to make a computer or other network device that is connected to it accessible to other computers and network devices from outside of the local network. Port forwarding uses a combination of an IP address and a port number to route network requests to specific devices. This technique is often used to make services on a host or gateway, residing on an internal network, accessible to a host on an external network by remapping the destination IP address and port number for the communication request.

Starting in Junos OS Release 18.3R1, TCP port forwarding (also referred to as TCP forwarding) enables a BNG to mediate communication between its connected access nodes and service provider back-office systems, such as external management and provisioning systems and TACACS+ servers. The BNG and its downstream access nodes are presented to back-office systems as a single addressable network element. You configure unique combinations of listening ports and addresses on the BNG. TCP connections are triggered when traffic from acceptable prefixes arrives on the listening port and matching listening address. Communication requests to and from access nodes are redirected from one address and port number combination to another when packets traverse the MX series router.

Back-office systems use NETCONF XML management protocol over SSH and TACACS+ to exchange requests with access nodes. For provisioning, they can use PCRF and RADIUS to supply service configurations for subscribers. Figure 28 on page 723 shows a sample topology for an external management system use case with optical line terminals (OLTs) connected to the BNG. Similar topologies might have different access nodes, such as DSLAMs, rather than OLTs.

Figure 28: Topology for Remote Device Management



The access nodes in this kind of topology act as logical extensions (remote devices) of the BNG so that the BNG can proxy all external management interactions for them. The BNG is configured with a public address and acts as the single point of management for itself and the access nodes. The remote devices have private addresses and are not publicly accessible. This means that the external systems cannot interact directly with the access nodes. The BNG must be able to mediate management requests between the access nodes and the management system, but it does not need to parse or act on the full content of the requests. This need is met with TCP port forwarding as follows for this use case:

- The external management system uses NETCONF XML protocol over SSH for tasks such as base configuration of the remote device before subscriber negotiation begins, configuration of Layer 2 data paths for new subscribers, displaying remote device status, and troubleshooting the remote device.

  In this case, the BNG demultiplexes requests from the management system to the remote devices.

- TACACS+ is used to authenticate and validate access to the remote device, perform system accounting, and control operator access.

  In this case, the BNG multiplexes request from the remote devices to the TACACS+ server that works with the external management system.

TCP port forwarding maps one or more combinations of an IPv4 listening address and a TCP port to destination addresses and ports so that the BNG can forward messages appropriately for both use cases. Each mapping is referred to as a *TCP connection pair*. TCP port forwarding operates as follows:

1. When the mapping is configured, the TCP port forwarding process opens the configured listening port and waits for an external system or access node to trigger a connection; that system or node can then be referred to as the *triggering entity*.

2. After the connection between the triggering entity and the BNG is established, TCP port forwarding attempts to open a TCP connection to the other half of the connection pair, which is the forwarding address and port combination defined in the mapping. TCP port forwarding examines only the TCP header information in the management traffic.

3. When both TCP connections have been established, TCP port forwarding monitors the connections for data traffic. When data is received on one connection, it is transmitted on the paired connection.

NOTE:

- If one side of the connection pair closes for any reason, TCP port forwarding closes the paired connection. This connection pair is not reestablished unless the triggering entity makes the connection on the TCP listening port again.

- If a configuration change is made to a TCP mapping while associated connection pairs are active, these connections are closed down. The connections are not reestablished unless the triggering entity makes the connection on the TCP listening port again

TCP port forwarding allows multiple simultaneous TCP connections for any single TCP mapping. You can place a limit on the maximum number of allowed connections.

You can use the following operational commands to manage and monitor TCP port forwarding:

- `clear tcp-forwarding connections`—Enables you to administratively close any current TCP connection pair.

- `clear tcp-forwarding statistics`—Enables you to clear (zero) statistics for the configured TCP mappings and any current TCP connection pairs. You can limit statistics clearing to all connections associated with a specific listening port/listening address combination or to only a single connection pair represented by a specific source address/source port combination. For either combination, you can optionally specify a routing instance; otherwise, the default routing instance is assumed.

- `show tcp-forwarding status`—Displays the status of TCP mapping and the current connections for each mapping. You can limit the display to a specific listening port/listening address combination, per routing instance. If you do not specify a routing instance, the default routing instance is assumed.

Traffic between the remote devices and the external systems is expected to be relatively small-sized management requests. Consequently, excessive traffic is not buffered and is dropped by TCP port

forwarding. TCP port forwarding does not maintain or recover established TCP connections in the event of a graceful Routing Engine switchover (GRES) or a daemon restart.

You can disable TCP port forwarding by including the `disable` statement at the `[edit system processes]` hierarchy level. You can also configure TCP port forwarding event tracing at the same hierarchy level by including the `traceoptions` statement. See "Tracing TCP Port Forwarding Events for Troubleshooting" on page 729 for more information.

Benefits of TCP Port Forwarding

- Simplifies BNG and remote device configuration and management in topologies that use external management and provisioning systems.
- TCP port forwarding is a generic functionality and can work with any application that can use TCP sessions for communication with remote devices and the BNG.
- Provides several options for tuning the TCP connections to your needs, including restriction to specific IPv4 prefixes, specific listening and forwarding address and port combinations, and the maximum number of allowed connections.

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
18.3R1	Starting in Junos OS Release 18.3R1, TCP port forwarding (also referred to as TCP forwarding) enables a BNG to mediate communication between its connected access nodes and service provider back-office systems, such as external management and provisioning systems and TACACS+ servers.

RELATED DOCUMENTATION

Configure TCP Port Forwarding for Remote Device Management | 725

Configure TCP Port Forwarding for Remote Device Management

To use TCP port forwarding, you configure the mapping between the TCP listening address/listening port combination on the BNG and the TCP port forwarding address/port combination where the BNG forwards the incoming data stream. TCP port forwarding is used when the BNG, together with one or more access nodes, is treated by an external management or provisioning system as a single addressable point of management. The remote devices have private addresses and are not publicly accessible. The

TCP port forwarding connections enable the BNG to demultiplex and multiplex management requests exchanged between the access nodes and the management system.

The listening port is monitored by the BNG for connections to be triggered by external management systems or a remote device. The listening address is a particular IPv4 address on the BNG that the triggering entity (external management/provisioning system or remote device) must use when attempting to trigger connections on the listening port.

By default, TCP connections are accepted from any source prefix. You can optionally configure one or more IPv4 prefixes from which TCP connections are accepted on the listening port. You can use a /32 IPv4 mask to specify a single address as the source or you can use other masks to specify an IPv4 subnet as the source. You can configure an unlimited number of prefixes for each listening port. To configure multiple prefixes, however, you must include the statement multiple times, once for each additional source prefix.

NOTE: Although not shown in the following steps, you can also configure TCP port forwarding in a non-default routing instance.

To configure a TCP mapping of a single TCP connection pair for TCP port forwarding:

1. Configure a unique combination of listening port and listening address for each TCP mapping.

```
[edit system services tcp-forwarding]
user@host# set listening-port port-number listening-address ipv4-listening-address
```

2. (Optional) Restrict the IPv4 prefixes from which TCP connections are accepted on the listening port. When you do not configure an allowed source, TCP connections are accepted from any source prefix.

```
[edit system services tcp-forwarding listening-port port-number listening-address ipv4-
listening-address]
user@host# set allowed-source ipv4-prefix
```

3. Define the IPv4 address to which MX BNG must open the second connection of the TCP pair after it opens the first connection triggered on the listening port/listening address combination. All packets received on one connection of the TCP pair are transmitted on the peer (second) connection. This address is used with the forwarding port to open the peer connection.

```
[edit system services tcp-forwarding listening-port port-number listening-address ipv4-
listening-address]
user@host# set forwarding-address ipv4-forwarding-address
```

4. Define the TCP port of the peer (second) connection of the TCP pair. This port is used with the forwarding address to open the peer connection.

```
[edit system services tcp-forwarding listening-port port-number listening-address ipv4-
listening-address]
user@host# set forwarding-port forwarding-port-number
```

5. (Optional) Set a limit on the number of simultaneous TCP connections that the BNG allows on a single listening port. Connection requests received after this limit is reached are rejected.

```
[edit system services tcp-forwarding listening-port port-number listening-address ipv4-
listening-address]
user@host# set max-connections number
```

> **NOTE**: In addition to this per-listening port limit, TCP port forwarding has a system-wide limit of 128 TCP connections (64 connection pairs) across all routing instances and listening ports.

The following sample configuration might be used for the topology shown in "TCP Port Forwarding for Remote Device Management" on page 722. In each step, the listening address is the public address of the BNG for management. A different listening port is assigned for the TACACS+ server, the management platform, and each remote device.

1. Configure the TACACS+ server connection. The BNG monitors port 8020 and its public address for TCP traffic from any of its remote devices to the TACACS server. It accepts traffic only from the subnet shared by the OLTs. It forwards acceptable traffic to the TACACS+ server on the IANA-assigned port number for TACACS, 49. The BNG supports four simultaneous TCP connections on the listening port/address combination, one for each OLT.

```
[edit system services tcp-forwarding]
user@host# edit listening-port 8020 listening-address 203.0.113.50
user@host# set allowed-source 192.0.0.1/24
user@host# set forwarding-address 198.51.100.1
user@host# set forwarding-port 49
user@host# set max-connections 4
```

2. Configure the NETCONF XML protocol connection to each remote device: OLT1, OLT2, OLT3, and OLT4. The BNG monitors its public address and four different ports for TCP traffic from the management platform to the remote devices. Each port is associated with one of the remote devices.

The BNG accepts traffic only from the management platform address, 198.51.100.3. Accepted traffic is forwarded to the associated device on the IANA-assigned port number for the NETCONF XML protocol over SSH, 830. Only one TCP connection is supported for each device.

a. Configure the NETCONF XML protocol connection to OLT1.

```
[edit system services tcp-forwarding]
user@host# edit listening-port 8000 listening-address 203.0.113.50
user@host# set allowed-source 198.51.100.3/32
user@host# set forwarding-address 192.0.0.2
user@host# set forwarding-port 830
user@host# set max-connections 1
```

b. Configure the NETCONF XML protocol connection to OLT2.

```
[edit system services tcp-forwarding]
user@host# edit listening-port 8001 listening-address 203.0.113.50
user@host# set allowed-source 198.51.100.3/32
user@host# set forwarding-address 192.0.0.3
user@host# set forwarding-port 830
user@host# set max-connections 1
```

c. Configure the NETCONF XML protocol connection to OLT3.

```
[edit system services tcp-forwarding]
user@host# edit listening-port 8002 listening-address 203.0.113.50
user@host# set allowed-source 198.51.100.3/32
user@host# set forwarding-address 192.0.0.4
user@host# set forwarding-port 830
user@host# set max-connections 1
```

d. Configure the NETCONF XML protocol connection to OLT4.

```
[edit system services tcp-forwarding]
user@host# edit listening-port 8003 listening-address 203.0.113.50
user@host# set allowed-source 198.51.100.3/32
user@host# set forwarding-address 192.0.0.5
user@host# set forwarding-port 830
user@host# set max-connections 1
```

RELATED DOCUMENTATION

TCP Port Forwarding for Remote Device Management | 722

Tracing TCP Port Forwarding Events for Troubleshooting

IN THIS SECTION

- Configuring the TCP Port Forwarding Trace Log Filename | 730
- Configuring the Number and Size of TCP Port Forwarding Log Files | 730
- Configuring Access to the TCP Port Forwarding Log File | 730
- Configuring a Regular Expression for TCP Port Forwarding Messages to Be Logged | 731
- Configuring the TCP Port Forwarding Tracing Flags | 731
- Configuring the Severity Level to Filter Which TCP Port Forwarding Messages Are Logged | 732

The Junos OS trace feature tracks TCP port forwarding operations and records events in a log file. The error descriptions captured in the log file provide detailed information to help you solve problems.

By default, nothing is traced. When you enable the tracing operation, the default tracing behavior is as follows:

1. Important events are logged in a file located in the `/var/log` directory. By default, the router uses the filename `tcpfwdd`. You can specify a different filename, but you cannot change the directory in which trace files are located.

2. When the trace log file *`filename`* reaches 128 kilobytes (KB), it is compressed and renamed *`filename`*`.0.gz`. Subsequent events are logged in a new file called *`filename`*, until it reaches capacity again. At this point, *`filename`*`.0.gz` is renamed *`filename`*`.1.gz` and *`filename`* is compressed and renamed *`filename`*`.0.gz`. This process repeats until the number of archived files reaches the maximum file number. Then the oldest trace file—the one with the highest number—is overwritten.

   You can optionally specify the number of trace files to be from 2 through 1000. You can also configure the maximum file size to be from 10 KB through 1 gigabyte (GB). (For more information about how log files are created, see the System Log Explorer.)

By default, only the user who configures the tracing operation can access log files. You can optionally configure read-only access for all users.

The following topics describe how to configure all aspects of tracing TCP port forwarding operations:

Configuring the TCP Port Forwarding Trace Log Filename

By default, the name of the file that records trace output for TCP port forwarding is `tcpfwdd`. You can specify a different name with the `file` option.

To configure the filename for TCP port forwarding tracing operations:

- Specify the name of the file used for the trace output.

```
[edit system processes tcp-forwarding traceoptions]
user@host# set file tcpfwd_1
```

Configuring the Number and Size of TCP Port Forwarding Log Files

You can optionally specify the number of compressed, archived trace log files to be from 2 through 1000. You can also configure the maximum file size to be from 10 KB through 1 gigabyte (GB); the default size is 128 kilobytes (KB).

The archived files are differentiated by a suffix in the format *.number*.gz. The newest archived file is .0.gz and the oldest archived file is *.(maximum number)-1*.gz. When the current trace log file reaches the maximum size, it is compressed and renamed, and any existing archived files are renamed. This process repeats until the maximum number of archived files is reached, at which point the oldest file is overwritten.

For example, you can 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 compressed and renamed *filename*.0.gz, and a new file called *filename* is created. When the new *filename* reaches 2 MB, *filename*.0.gz is renamed *filename*.1.gz and *filename* is compressed and renamed *filename*.0.gz. This process repeats until there are 20 trace files. Then the oldest file, *filename*.19.gz, is simply overwritten when the next oldest file, *filename*.18.gz is compressed and renamed to *filename*.19.gz.

To configure the number and size of trace files:

- Specify the name, number, and size of the file used for the trace output.

```
[edit system processes tcp-forwarding traceoptions]
user@host# set file tcpfwd_1 _logfile_1 files 20 size 2097152
```

Configuring Access to the TCP Port Forwarding Log File

By default, only the user who configures the tracing operation can access the log files. You can enable all users to read the log file and you can explicitly set the default behavior of the log file.

To specify that all users can read the log file:

- Configure the log file to be world-readable.

```
[edit system processes tcp-forwarding traceoptions]
user@host# set file tcpfwd_1 _logfile_1 world-readable
```

To explicitly set the default behavior, only the user who configured tracing can read the log file:

- Configure the log file to be no-world-readable.

```
[edit system processes tcp-forwarding traceoptions]
user@host# set file tcpfwd_1 _logfile_1 no-world-readable
```

Configuring a Regular Expression for TCP Port Forwarding Messages to Be Logged

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

You can refine the output by including regular expressions to be matched.

To configure regular expressions to be matched:

- Configure the regular expression.

```
[edit system processes tcp-forwarding traceoptions]
user@host# set file tcpfwd_1 _logfile_1 match regex
```

Configuring the TCP Port Forwarding Tracing Flags

By default, only important events are logged. You can specify which events and operations are logged by specifying one or more tracing flags.

To configure the flags for the events to be logged:

- Configure the flags.

```
[edit system processes tcp-forwarding traceoptions]
user@host# set flag flag-name
```

Configuring the Severity Level to Filter Which TCP Port Forwarding Messages Are Logged

The messages associated with a logged event are categorized according to severity level. You can use the severity level to determine which messages are logged for the event type. The severity level that you configure depends on the issue that you are trying to resolve. In some cases you might be interested in seeing all messages relevant to the logged event, so you specify `all` or `verbose`. Either choice generates a large amount of output. You can specify a more restrictive severity level, such as `notice` or `info` to filter the messages . By default, the trace operation output includes only messages with a severity level of `error`.

To configure the type of messages to be logged:

- Configure the message severity level.

```
[edit system processes tcp-forwarding traceoptions]
user@host# set level severity
```

RELATED DOCUMENTATION

TCP Port Forwarding for Remote Device Management | 722

CHAPTER 47

Configuring IPFIX Mediation for Remote Device Monitoring

IN THIS CHAPTER

- IPFIX Mediation on the BNG | 733
- Configuring the BNG as an IPFIX Mediator to Collect and Export IPFIX Data | 739

IPFIX Mediation on the BNG

IN THIS SECTION

- Template ID Reconciliation | 735
- IPFIX Mediation and Network Analytics | 737
- Benefits of IPFIX Mediation | 738

Traffic flow is a way of conceptualizing how IP data traffic passes through the various components of your network. A flow consists of a set of IP packets that pass an observation point in the network during a specific time interval. The set is defined by common properties:

- One or more packet, transport, or application header fields
- One or more characteristics of the packet
- One or more fields derived from how the packet is handled

For example, a particular flow might include packets with the same destination IP address and destination port number, number of MPLS labels, next-hop address, and output interface.

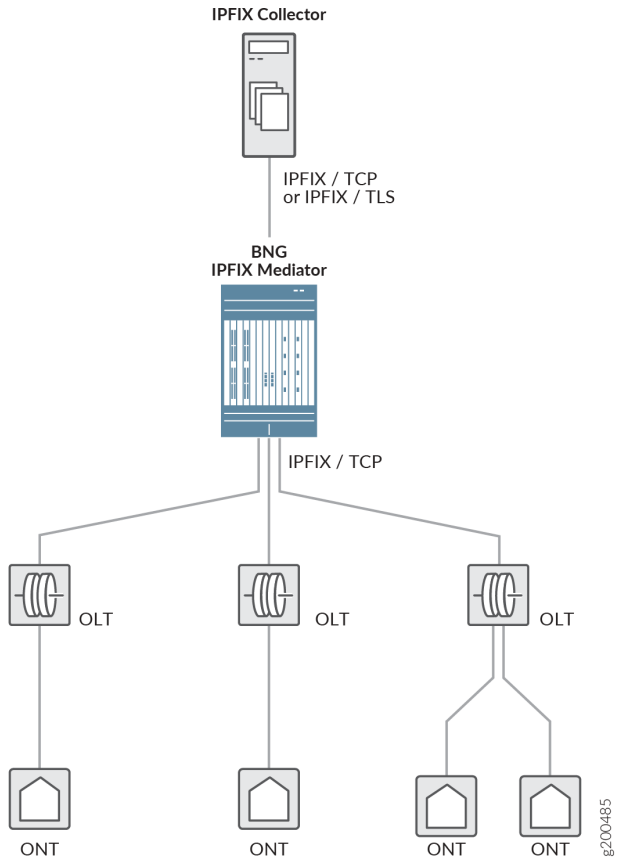
The IP Flow Information Export (IPFIX) protocol is a mechanism for transmitting traffic flow information in the form of flow records over your network from an exporting process to a collecting process. Each

flow record is generated by a monitoring process and contains information about a specific flow at the observation point, such as the total number of bytes for all packets in the flow and the source IP address. The device that hosts one or more exporting processes is called an exporter or IPFIX device. The device that receives (collects) the flow records from one or more exporting processes is called the collector.

Starting in Junos OS Release 18.3R1, you can configure an MX Series router acting as a BNG to be an intermediary device between IPFIX exporters and collectors. As an IPFIX mediator, the BNG functions as both a collector and an exporter. The IPFIX mediator function collects performance management data via IPFIX records from downstream access network devices such as OLTs and advanced ONUs (with integrated functions such as IPFIX exporter, VOIP SIP client, and so on). This data along with local performance management data from the MX BNG is aggregated and relayed to an upstream IPFIX collector. From the reference point of the IPFIX collector, IPFIX mediation enables the router and its associated access network devices to appear as a single IPFIX export source leveraging a single TCP/IP connection between the MX BNG and the upstream collector.

Figure 29 on page 735 shows a Passive Optical Network (PON) topology where the BNG IPFIX mediator connected to downstream OLTs, which are in turn connected further downstream to ONTs in residences. The downstream devices export flow information to the mediator over TCP/IP connections; the mediator collects the flow information from the downstream devices. The mediator then processes the flow information and exports it upstream to the IPFIX collector over a TCP or Transport Layer Security (TLS) connection.

Figure 29: Sample Network Topology for IPFIX Mediation



The IPFIX Mediator function enables the BNG and its associated downstream devices to be represented as a single IPFIX exporter towards the IPFIX collector. The data records are not formatted, which optimizes the efficiency of the data stream. A template record, sometimes referred to simply as a template, specifies the semantics and structure for a flow record as an ordered sequence of <type, length> pairs. Template records are sent either before the data records or inline with them.

Each template record includes the template header and one or more field specifiers corresponding to information elements in the data records. The template header includes the template ID and a count of the fields in the template record. The template ID is unique to the transport session and observation domain (where the traffic flow was observed). Effectively, the ID is unique to the TCP connection between a downstream exporting device and the mediator. Consequently, different downstream devices are likely to use different template IDs for the same record type.

Template ID Reconciliation

One aspect of the mediation processing is template ID reconciliation. The mediator maintains a cache of unique template records received from the downstream exporters. Matching template records received from different export sources are mapped to the same instance of the record in the template cache. The

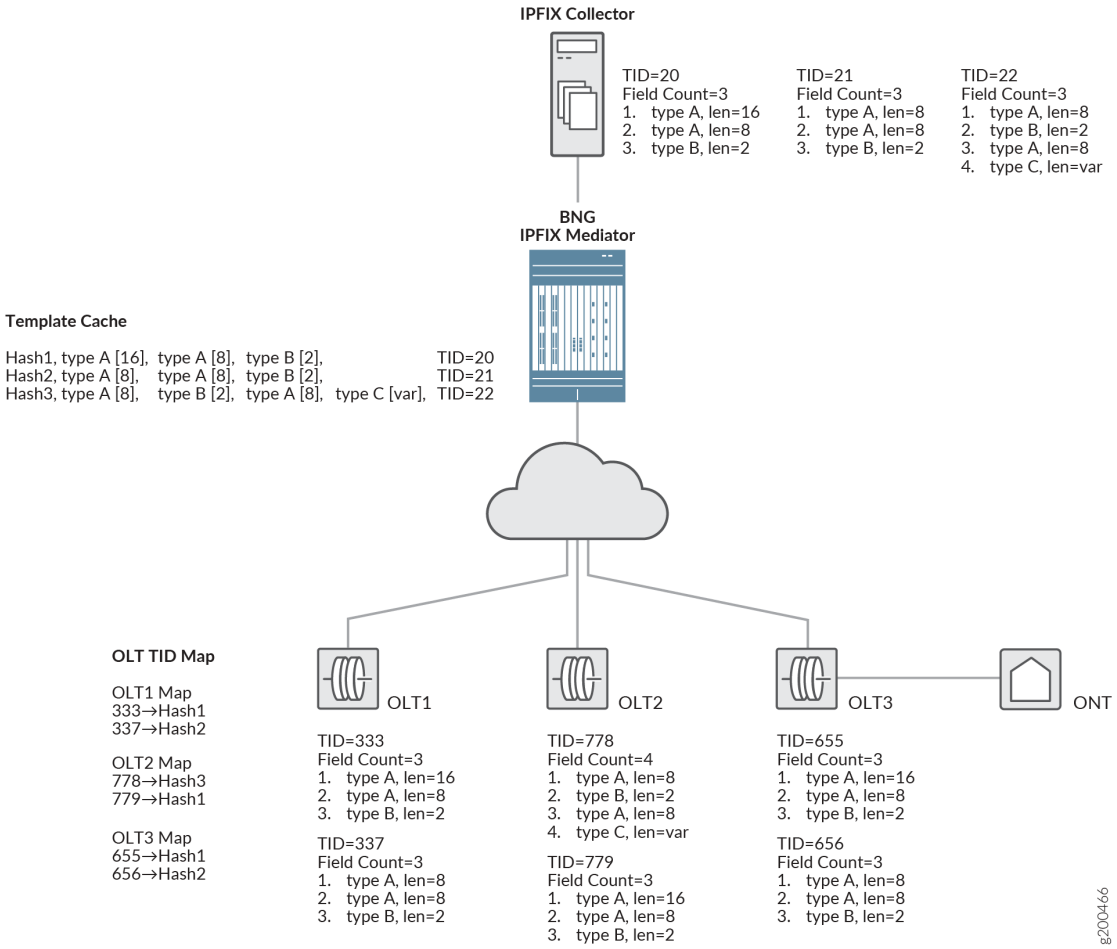
incoming template records are matched according to the hash value of the number, type, length, and order of the record fields. In other words, the mediator uniquely identifies template records independent of their IDs.

This enables the mediator to assign a new template ID for each unique received template ID. The new upstream ID is used for exporting the template record and data records to the upstream collector. Each new ID is unique to the transport session (TCP or TLS) between the mediator and the collector. This processing results in significantly streamlined communication between the mediator and the collector compared to sending records separately that match except for their template IDs.

Figure 30 on page 737 shows how reconciliation works.

1. The IPFIX mediator receives two template records with different IDs from each OLT.
2. By comparing the hash value for the number and order of the fields and the type and length values for each field, the mediator determines that the six template records from the OLTs represent only three unique records, as follows:
   - The template records with IDs of 333 (OLT1), 779, (OLT2), and 655 (OLT3) all have the same hash value and consequently describe the same record.
   - The template records with IDs of 337 (OLT1) and 656 (OLT3) both have the same hash value and consequently describe the same record.
   - The template record with ID of 778 (OLT2) has a hash value that does not match any other records.
3. Each unique template record is stored in the template cache and assigned a new template ID that is used for sending template and data records to the collector.

Figure 30: Template ID Reconciliation



NOTE: If the IPFIX mediator receives any data records without receiving a corresponding template record in the same TCP session, it discards the data records and logs the event.

The IPFIX mediator functions in a pass-through capacity for the data records from the downstream devices. It does not modify the data records other than changing the template ID for export to the collector. The mediator does not differentiate the data received from different downstream devices; that function is left to the IPFIX collector.

IPFIX Mediation and Network Analytics

IPFIX mediation on the MX Series router employs plug-ins for the `ipfix` network analytics service agent to receive, process, and export IPFIX records. The input plug-in (`input-ipfix`) listens for IPFIX messages on TCP connections from the downstream exporting devices, using port 4739 by default. No other

message types are expected or accepted. The output plug-in (`output-ipfix`) reconciles the received records and sends them to the destination IPFIX collector, which is assumed to listen for them on TCP port 4740 by default. Both plug-ins enable you to configure different parameters for IPFIX mediation. For example, whether the mediator attempts a TLS or TCP connection to the collector is determined by the configuration of certificate options in the output plug-in.

NOTE: The IPFIX plug-ins work only with each other and not with any other analytics plug-in.

Benefits of IPFIX Mediation

- An IPFIX mediator reduces the load on the collector without a loss of information. As the amount of traffic grows in a specific network, the capacity of a single collector to process flow records from multiple exporters can be exceeded. Packet sampling and aggregation can reduce the amount of data to be processed, at the risk of the potential loss of small flows and the detailed information that might be needed to detect and deal with some traffic changes and anomalous behavior.
- An IPFIX mediator provides the flexibility needed when you use multiple traffic monitoring applications. Different applications may require different levels of information, such as packet level versus flow level. These different needs might force the exporter to run different metering tasks to generate flow records, straining limited resources on the device.
- An IPFIX mediator simplifies the accurate monitoring, processing, and exporting of information in networks with a variety of IPFIX devices from multiple vendors, running multiple software releases. A single BNG can mediate the differences between many connected IPFIX devices before exporting flow records to the collector, removing that burden from the individual collectors.

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
18.3R1	Starting in Junos OS Release 18.3R1, you can configure an MX Series router acting as a BNG to be an intermediary device between IPFIX exporters and collectors.

RELATED DOCUMENTATION

Configuring the BNG as an IPFIX Mediator to Collect and Export IPFIX Data | 739

Configuring the BNG as an IPFIX Mediator to Collect and Export IPFIX Data

IPFIX mediation uses the `ipfix` analytics service agent. The service agent uses input and output plug-ins specific to IPFIX. The plug-ins configure aspects of the collecting and exporting functions for the mediator, such as TCP ports and the collector address. The input plug-in takes in the IPFIX flow data from the downstream devices. The output plug-in converts the data to the IPFIX format and exports it to the IPFIX collector. Data conversion is particularly important because users may have a variety of exporting devices using different formats. Converting the formats to a common form on the mediator alleviates the need to have specific collectors for different formats.

Your configuration for the output plug-in determines whether the IPFIX mediator sends records to the collector over a TCP connection or a TLS connection:

- When you configure any of the certificate options (`collector-ca-certificate`, `collector-certificate-key`, or `collector-certificate`), the mediator attempts to make a TLS connection.
- If none of the certificate options is configured, the mediator attempts to make a TCP connection.

To configure IPFIX mediation:

1. Access the IPFIX service agent configuration.

```
[edit services analytics agent]
user@host# edit service-agents ipfix
```

2. Configure parameters for the IPFIX input plug-in.

```
[edit services analytics agent service-agents ipfix]
user@host# edit inputs input-ipfix
```

NOTE: Although each of the parameters has a default value, you must configure at least one of the parameters to enable the plug-in. If you configure only one parameter and want to use the default value, you must specify that value.

a. (Optional) Specify the maximum number of TCP connections that the IPFIX mediator can have. The default value is 100.

```
[edit services analytics agent service-agents ipfix inputs input-ipfix]
user@host# set parameters maximum-connections number
```

b. (Optional) Specify the TCP port that the IPFIX mediator uses to receive TCP packets from the downstream devices. The default value is 4739.

```
[edit services analytics agent service-agents ipfix inputs input-ipfix]
user@host# set parameters tcp-port port-number
```

c. Specify the name of the VRF (routing instance) where IPFIX packets are accepted from the downstream devices.

```
[edit services analytics agent service-agents ipfix inputs input-ipfix]
user@host# set parameters vrf-name name
```

3. Configure parameters for the output plug-in.

```
[edit services analytics agent service-agents ipfix]
user@host# edit outputs output-ipfix
```

a. Specify the IP address of the upstream IPFIX collector. This is a mandatory option.

```
[edit services analytics agent service-agents ipfix outputs output-ipfix]
user@host# set parameters collector-address ip-address
```

b. (Optional) Specify the path for the certificate that is used to sign the peer certificate at the peer (IPFIX collector) level. The certificate is provided by a trusted certificate authority (CA) and is expected to be in .pem container format.

```
[edit services analytics agent service-agents ipfix outputs output-ipfix]
user@host# set parameters collector-ca-certificate file-path
```

c. (Optional) Specify the path for the client certificate that the server (IPFIX collector) uses to authenticate the client and to enable mutual authentication. The fully-qualified domain name

(FQDN) of both the client and the server are stored in the certificate's Subject Alternative Name field when the client and server certificates are generated. The certificate is expected to be in .pem container format.

```
[edit services analytics agent service-agents ipfix outputs output-ipfix]
user@host# set parameters collector-certificate file-path
```

d. (Optional) Specify the path of the private key file that is loaded to decrypt the encrypted message sent from the peer.

```
[edit services analytics agent service-agents ipfix outputs output-ipfix]
user@host# set parameters collector-certificate-key file-path
```

e. (Optional) Specify how many seconds the output plug-in waits before retrying the connection to the IPFIX collector. The default value is 20.

```
[edit services analytics agent service-agents ipfix outputs output-ipfix]
user@host# set parameters collector-connection-retry-interval seconds
```

f. (Optional) Specify the TCP port that the IPFIX mediator uses to connect to the IPFIX collector. The default value is 4740.

```
[edit services analytics agent service-agents ipfix outputs output-ipfix]
user@host# set parameters collector-tcp-port port-number
```

g. (Optional) Specify the name of the VRF (routing instance) in which IPFIX packets are routed to the IPFIX collector. The default value is `default`.

```
[edit services analytics agent service-agents ipfix outputs output-ipfix]
user@host# set parameters collector-vrf-name vrf-name
```

In the following sample configuration, the input plug-in is configured so that the IPFIX mediator accepts up to 125 TCP connections from its downstream devices. Records are accepted in the RI-ipfix-1 routing instance. The TCP port is not configured, so the plug-in listens on the default port, 4739.

```
[edit services analytics agent service-agents ipfix]
user@host# set inputs input-ipfix parameters maximum-connections 125
user@host# set inputs input-ipfix parameters vrf-name RI-ipfix-1
```

The following example configuration for the output plug-in specifies that:

- Records are exported to the collector at 198.51.100.200.
- If the connection to the collector is not successful, the plug-in attempts to make the connection at 15-second intervals.
- The configuration includes paths for collector certificates, so the export connection is over TLS rather than TCP.
- The TCP port is not configured, so the collector is expected to listen on the default port, 4740.
- No routing instance is configured for the collector, so it accepts packets in the default routing instance.

```
user@host# edit services analytics agent service-agents ipfix
user@host# set outputs output-ipfix parameters collector-address 198.51.100.200
user@host# set outputs output-ipfix parameters collector-ca-certificate /var/tmp/ca.pem
user@host# set outputs output-ipfix parameters collector-certificate /var/tmp/client.pem
user@host# set outputs output-ipfix parameters collector-certificate-key /var/tmp/example.com.key
user@host# set outputs output-ipfix parameters collector-connection-retry-interval 15
```

RELATED DOCUMENTATION

IPFIX Mediation on the BNG | 733

CHAPTER 48

Collection and Export of Local Telemetry Data on the IPFIX Mediator

IN THIS CHAPTER

- Telemetry Data Collection on the IPFIX Mediator for Export to an IPFIX Collector | 743
- Configuring the Collection and Export of Local Telemetry Data on the IPFIX Mediator | 747

Telemetry Data Collection on the IPFIX Mediator for Export to an IPFIX Collector

IN THIS SECTION

- Benefits of Telemetry Data Collection | 746

Starting in Junos OS Release 18.4R1, the input-jti-ipfix plug-in for the IPFIX service agent collects telemetry data from the local Junos Telemetry Interface (JTI) on a BNG configured as an IPFIX mediator. The output-ipfix plug-in translates the gRPC data received from a nonconfigurable set of well-defined sensor types into specific IPFIX records. You configure a record group for the input plug-in that consists of one or more of the predefined IPFIX records. Each record is associated with a specific set of telemetry sensors on the BNG, as listed in Table 54 on page 744.

Table 54: IPFIX Records and Associated Telemetry Sensors (gRPC Path)

IPFIX Record Name	Sensors Collected by the Record
address-pool-utilization	/junos/system/subscriber-management/aaa/address-assignment-statistics/logical-system-routing-instances/logical-system-routing-instance/pools/pool/
chassis-inventory	/components/component[name='Routing EngineX']/properties/property[name='fru-model-number'] /components/component[name='FPCx']/properties/property[name='fru-model-number'] /components/component[name='Routing Engine0']/state/id /components/component[name='FPCx']/state/id /components/component[name='Routing EngineX']/properties/property[name='hardware-rev']/ /components/component[name='FPCx']/properties/property[name='hardware-rev']/ /components/component[name='FPCx']/properties/property[name='software-rev']/
chassis-power	/components/component[name='Chassis']/properties/property[name='power-system-capacity']/ /components/component[name='Chassis']/properties/property[name='power-system-remaining']/
dhcpv4-server-statistics	/junos/system/subscriber-management/client-protocols/dhcp/v4/routing-instances/routing-instance/server/statistics/
interface-meta-data	/junos/system/subscriber-management/dynamic-interfaces/interfaces/meta-data/interface
interface-queue-statistics	/junos/system/subscriber-management/dynamic-interfaces/interfaces/queue-statistics/interface
port-statistics	/interfaces/interface/state/counters/

Table 54: IPFIX Records and Associated Telemetry Sensors (gRPC Path) *(Continued)*

IPFIX Record Name	Sensors Collected by the Record
resource-utilization	/components/component[name='Routing Engine0']/properties/property[name='memory-dram-used']/ /components/component[name='Routing Engine1']/properties/property[name='memory-dram-used']/ /components/component[name='Routing Engine0']/properties/property[name='memory-dram-installed']/ /components/component[name='Routing Engine1']/properties/property[name='memory-dram-installed']/ /components/component[name='FPCx']/properties/property[name='memory-utilization-heap']/ /components/component[name='Routing Engine0']/properties/property[name='memory-utilization-buffer']/ /components/component[name='Routing Engine1']/properties/property[name='memory-utilization-buffer']/ /components/component[name='FPCx']/properties/property[name='memory-utilization-buffer']/ /components/component[name='Routing Engine0']/properties/property[name='cpu-utilization-idle']/ /components/component[name='Routing Engine1']/properties/property[name='cpu-utilization-idle']/
subscriber-statistics	/junos/system/subscriber-management/dynamic-interfaces/interfaces/subscriber-statistics/interface
thermal	/components/component[name='Chassis']/properties/property[name='temperature-ambient'] /components/component[name='RoutingEngine0']/properties/property[name='temperature']/state/value /components/component[name='RoutingEngine1']/properties/property[name='temperature']/state/value /components/component[name=FPCx]/properties/property[name='temperature-exhaust-x']/state/value

Table 54: IPFIX Records and Associated Telemetry Sensors (gRPC Path) *(Continued)*

IPFIX Record Name	Sensors Collected by the Record
uptime	/components/component[name='FPCx']/properties/property[name='uptime']/

BEST PRACTICE: We recommend that you configure the `interface-metadata` record whenever you configure the `interface-queue-statistics` record. The metadata information is essential for understanding details about the subscriber whose queue statistics are being collected.

You can configure the frequency with which data is collected and reported to an IPFIX collector. The reporting interval has a default value, but some telemetry data, such as subscriber statistics, is more dynamic than other data, such as chassis temperature. A shorter reporting interval may be more useful for the more dynamic data. You can configure the reporting interval for the record group, but not for individual records.

The template IDs for the translated gRPC data are drawn from the same template ID space as the IPFIX mediator. Template IDs in the range 256 though 400 are reserved for the translation of telemetry data.

NOTE:
For detailed information about the Juniper Telemetry Interface, see Junos Telemetry Interface User Guide.

Benefits of Telemetry Data Collection

- Leverages the IPFIX mediation structure to collect data about hardware, resources, and user statistics for better remote management of the BNG and subscribers.

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, the input-jti-ipfix plug-in for the IPFIX service agent collects telemetry data from the local Junos Telemetry Interface (JTI) on a BNG configured as an IPFIX mediator.

RELATED DOCUMENTATION

Configuring the Collection and Export of Local Telemetry Data on the IPFIX Mediator | 747

Configuring the BNG as an IPFIX Mediator to Collect and Export IPFIX Data | 739

IPFIX Mediation on the BNG | 733

Enabling Export of Subscriber Statistics and Queue Statistics for Dynamic Interfaces and Interface-Sets

Configuring the Collection and Export of Local Telemetry Data on the IPFIX Mediator

You can configure the input-jti-ipfix plug-in for the IPFIX service agent to collect telemetry (gRPC) data from the local Junos Telemetry Interface (JTI) on a BNG configured as an IPFIX mediator. In addition to streaming IPFIX records from the input-ipfix plug-in, the output-ipfix plug-in also translates the gRPC data received from the input-jti-ipfix plug-into corresponding IPFIX data records.

You configure a record group for the input-jti-ipfix plug-in that consists of one or more predefined IPFIX records. Each predefined record is associated with a specific, nonconfigurable set of telemetry sensors on the BNG. You can configure the frequency at which the sensor records are exported to an IPFIX collector; the IPFIX collector is configured with the output-ipfix plug-in.

Before you begin, you must enable the IPFIX service agent by configuring at least one parameter for the input-ipfix plug-in.

To configure local telemetry data collection and reporting:

1. Access the IPFIX service agent configuration.

```
[edit services analytics agent]
user@host# edit service-agents ipfix
```

2. Configure parameters for the IPFIX telemetry input plug-in.

```
[edit services analytics agent service-agents ipfix]
user@host# edit inputs input-jti-ipfix
```

a. Specify the name of a group of records to collect telemetry data.

```
[edit services analytics agent service-agents ipfix inputs input-jti-ipfix]
user@host# edit parameters record-group group-name
```

b. Specify the record that you want to add to the group.

```
[edit services analytics agent service-agents ipfix inputs input-jti-ipfix parameters
record-group group-name
user@host# set record ipfix-record-name
```

c. (Optional) Configure a reporting interval for the record group when you do not want to use the default value (900 seconds).

```
[edit services analytics agent service-agents ipfix inputs input-jti-ipfix parameters
record-group group-name]
user@host# set reporting-interval seconds
```

3. Configure parameters for the IPFIX output plug-in. This is the same configuration you use when you configure the IPFIX mediation.

```
[edit services analytics agent service-agents ipfix]
user@host# edit outputs output-ipfix
```

a. Specify the IP address of the upstream IPFIX collector. This is a mandatory option.

```
[edit services analytics agent service-agents ipfix outputs output-ipfix]
user@host# set parameters collector-address ip-address
```

b. (Optional) Specify the path for the certificate that is used to sign the peer certificate at the peer (IPFIX collector) level. The certificate is provided by a trusted certificate authority (CA) and is expected to be in .pem container format.

```
[edit services analytics agent service-agents ipfix outputs output-ipfix]
user@host# set parameters collector-ca-certificate file-path
```

c. (Optional) Specify the path for the client certificate that the server (IPFIX collector) uses to authenticate the client and to enable mutual authentication. The fully-qualified domain name (FQDN) of both the client and the server are stored in the certificate's Subject Alternative Name field when the client and server certificates are generated. The certificate is expected to be in .pem container format.

```
[edit services analytics agent service-agents ipfix outputs output-ipfix]
user@host# set parameters collector-certificate file-path
```

d. (Optional) Specify the path of the private key file that is loaded to decrypt the encrypted message sent from the peer.

```
[edit services analytics agent service-agents ipfix outputs output-ipfix]
user@host# set parameters collector-certificate-key file-path
```

e. (Optional) Specify how many seconds the output plug-in waits before retrying the connection to the IPFIX collector. The default value is 20.

```
[edit services analytics agent service-agents ipfix outputs output-ipfix]
user@host# set parameters collector-connection-retry-interval seconds
```

f. (Optional) Specify the TCP port that the IPFIX mediator uses to connect to the IPFIX collector. The default value is 4740.

```
[edit services analytics agent service-agents ipfix outputs output-ipfix]
user@host# set parameters collector-tcp-port port-number
```

g. (Optional) Specify the name of the VRF (routing instance) in which IPFIX packets are routed to the IPFIX collector. The default value is `default`.

```
[edit services analytics agent service-agents ipfix outputs output-ipfix]
user@host# set parameters collector-vrf-name vrf-name
```

The following sample configuration includes three record groups for the telemetry input plug-in, high-frequency, baseline, and background:

- The high-frequency group subscribers to the subscriber-statistics and port-statistics record. Because statistics data is dynamic and changes frequently, the reporting interval is set to five minutes, which is much less than the default.
- The baseline record group subscribes to the address-pool-utilization and dhcpv4-server-statistics records; the reporting interval is left at the default value, 15 minutes.
- The background record group subscribes to the thermal and chassis-inventory records. These probably do not change frequently, so the reporting interval is set to six hours.

To enable the IPFIX plug-in, you must configure at least one parameter; in this example, the maximum number of TCP connections is set to 200.

Finally, the IP address and listening port for the IPFIX collector is configured in the output plug-in.

```
[edit services analytics agent service-agents ipfix]
    inputs input-jti-ipfix {
        parameters {
            record-group high-frequency {
                record subscriber-statistics;
                record port-statistics;
                reporting-interval 300;
            }
            record-group baseline {
                record address-pool-utilization;
                record dhcpv4-server-statistics;
            }
            record-group background {
                record thermal;
                record chassis-inventory;
                reporting-interval 21600;
            }
        }
    }
    inputs input-ipfix {
        parameters {
            maximum-connections 200;
        }
    }
    outputs output-ipfix {
        parameters {
            collector-address 192.0.2.2;
            collector-port 6589;
```

```
        }
    }
}
```

You can use the *show services analytics agent* command to display information about the service agents.

RELATED DOCUMENTATION

- Telemetry Data Collection on the IPFIX Mediator for Export to an IPFIX Collector | **743**
- Configuring the BNG as an IPFIX Mediator to Collect and Export IPFIX Data | **739**
- IPFIX Mediation on the BNG | **733**

PART 10

Troubleshooting

- Contacting Juniper Networks Technical Support | **753**
- Knowledge Base | **756**

CHAPTER 49

Contacting Juniper Networks Technical Support

IN THIS CHAPTER

- Collecting Subscriber Access Logs Before Contacting Juniper Networks Technical Support | 753

Collecting Subscriber Access Logs Before Contacting Juniper Networks Technical Support

IN THIS SECTION

- Problem | 753
- Solution | 753

Problem

Description

When you experience a subscriber access problem in your network, we recommend that you collect certain logs before you contact Juniper Networks Technical Support. This topic shows you the most useful logs for a variety of network implementations. In addition to the relevant log information, you must also collect standard troubleshooting information and send it to Juniper Networks Technical Support in your request for assistance.

Solution

To collect standard troubleshooting information:

- Redirect the command output to a file.

```
user@host> request support information | save rsi-1
```

To configure logging to assist Juniper Networks Technical Support:

1. Review the following blocks of statements to determine which apply to your configuration.

```
[edit]
set system syslog archive size 100m files 25
set system auto-configuration traceoptions file filename
set system auto-configuration traceoptions file filename size 100m files  25
set protocols ppp-service traceoptions file filename size 100m files  25
set protocols ppp-service traceoptions level all
set protocols ppp-service traceoptions flag all
set protocols ppp traceoptions file filename size 100m files 25
set protocols ppp traceoptions level all
set protocols ppp traceoptions flag all
set protocols ppp monitor-session all
set interfaces pp0 traceoptions flag all
set demux traceoptions file filename size 100m files 25
set demux traceoptions level all
set demux traceoptions flag all
set system processes dhcp-service traceoptions file filename
set system processes dhcp-service traceoptions file size 100m
set system processes dhcp-service traceoptions file files 25
set system processes dhcp-service traceoptions flag all
set class-of-service traceoptions file filename
set class-of-service traceoptions file size 100m
set class-of-service traceoptions flag all
set class-of-service traceoptions file files 25
set routing-options traceoptions file filename
set routing-options traceoptions file size 100m
set routing-options traceoptions flag all
set routing-options traceoptions file files 25
set interfaces traceoptions file filename
set interfaces traceoptions file size 100m
set interfaces traceoptions flag all
set interfaces traceoptions file files 25
set system processes general-authentication-service traceoptions file filename
set system processes general-authentication-service traceoptions file size 100m
```

```
set system processes general-authentication-service traceoptions flag all
set system processes general-authentication-service traceoptions file files 25
```

2. Copy the relevant statements into a text file and modify the log filenames as you want.
3. Copy the statements from the text file and paste them into the CLI on your router to configure logging.
4. Commit the logging configuration to begin collecting information.

NOTE: The maximum file size for DHCP local server and DHCP relay log files is 1 GB. The maximum number of log files for DHCP local server and DHCP relay is 1000.

BEST PRACTICE: Enable these logs only to collect information when troubleshooting specific problems. Enabling these logs during normal operations can result in reduced system performance.

RELATED DOCUMENTATION

- Compressing Troubleshooting Logs from /var/logs to Send to Juniper Networks Technical Support

CHAPTER 50

Knowledge Base

PART 11

Configuration Statements and Operational Commands

- Junos CLI Reference Overview | 758

Junos CLI Reference Overview

We've consolidated all Junos CLI commands and configuration statements in one place. Read this guide to learn about the syntax and options that make up the statements and commands. Also 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