JUNIPER NETWORKS

Next-Generation Network Addressing Carrier-Grade NAT and IPv6 Solutions

Release
12.1

Published: 2012-04-17

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Next-Generation Network Addressing Carrier-Grade NAT and IPv6 Solutions
12.1

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 http://www.juniper.net/support/eula.html. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

About the Documentation xiii
Documentation and Release Notes xiii
Supported Platforms xiii
Using the Examples in This Manual xiii
Merging a Full Example xiv
Merging a Snippet xiv
Documentation Conventions xv
Documentation Feedback xvii
Requesting Technical Support xvii
Self-Help Online Tools and Resources xvii
Opening a Case with JTAC xviii

Part 1 Overview

Chapter 1 Next-Generation Network Addressing 3

Next-Generation Network Addressing Overview 3
Sample IPv6 Transition Scenarios 3
Example 1: IPv4 Depletion with a Non-IPv6 Access Network 3
Example 2: IPv4 Depletion with an IPv6 Access Network 4
Example 3: IPv4 Depletion for Mobile Networks 5

Chapter 2 Carrier-Grade NAT Solutions 7

Network Address Translation Overview 7
Types of NAT 7
NAT Concept and Facilities Overview 7
IPv4-to-IPv4 Basic NAT 8
NAT-PT 9
Static Destination NAT 9
Twice NAT 9
IPv6 NAT 10
NAT-PT with DNS ALG 10
Dynamic NAT 10
Stateful NAT64 11
Dual-Stack Lite 11

Chapter 3 Tunneling Solutions 13

Tunneling Services for IPv4-to-IPv6 Transition Overview 13
6to4 Overview 13
Basic 6to4 14
6to4 Anycast 14
6to4 Provider-Managed Tunnels 15
DS-Lite Softwires—IPv4 over IPv6 15

6rd Softwires—IPv6 over IPv4 16

Part 2 Configuration

Chapter 4 NAT Configuration Tasks 21

Configuring Addresses and Ports for Use in NAT Rules 21
Configuring Pools of Addresses and Ports 22
Preserve Range and Preserve Parity 22
Configuring Address Pools for Network Address Port Translation 23
Round-Robin Allocation 23
Sequential 24
Port Block Allocation 25
Additional Options for NAPT 28
Comparision of NAPT Implementation Methods 28
Specifying Destination and Source Prefixes 29
Requirements for NAT Addresses 29
Configuring NAT Rules 30
Configuring Match Direction for NAT Rules 31
Configuring Match Conditions in NAT Rules 32
Configuring Actions in NAT Rules 33
Configuring NAT Rule Sets 35
Configuring NAT Service Sets 36
Configuring Static Source Translation in IPv4 Networks 37
Configuring the NAT Pool and Rule 38
Configuring the Service Set for NAT 39
Configuring Trace Options 40
Configuring Static Source Translation in IPv6 Networks 41
Configuring the NAT Pool and Rule 41
Configuring the Service Set for NAT 42
Configuring Trace Options 43
Configuring Dynamic Source Address and Port Translation in IPv4 Networks . . . 44
Configuring Dynamic Source Address and Port Translation for IPv6 Networks 47
Configuring Dynamic Address-Only Source Translation in IPv4 Networks 48
Configuring Static Destination Address Translation in IPv4 Networks 51
Configuring Translation Type for Translation Between IPv6 and IPv4 Networks 53
Configuring the DNS ALG Application 53
Configuring the NAT Pool and NAT Rule 54
Configuring the Service Set for NAT 57
Configuring Trace Options 58
Configuring NAT-PT 58
Example: Configuring Port Forwarding with Twice NAT 60
Configuring NAT-PT 62
Configuring Dynamic Source Address and Static Destination Address Translation (IPv6 to IPV4) 65
Configuring Port Forwarding for Static Destination Address Translation 66
Configuring Port Forwarding Without Destination Address Translation 69
Configuring Secured Port Block Allocation 70

Configuring Deterministic Port Block Allocation 72

Chapter 5 NAT Rules Examples 75

Example: Configuring Static Source Translation in an IPv4 Network 76
Example: Configuring Static Source Translation in an IPv6 Network 76
Example: Configuring Static Source Translation with Multiple Prefixes and Address Ranges 77
Example: Configuring Dynamic Source Address and Port Translation (NAPT) for an IPv4 Network 78
Example: Configuring Dynamic Source Translation for an IPv4 Network 78
Example: Configuring Dynamic Address-Only Source Translation 79
Example: Configuring Dynamic Address-Only Source Translation in an IPv4 Network 79
Example: Configuring Static Destination Address Translation 80
Example: Configuring Dynamic Source Address and Static Destination Address Translation (IPv6 to IPV4) 81
Example: Configuring NAT in Mixed IPv4 and IPv6 Networks 81
Example: Configuring the Translation Type Between IPv6 and IPv4 Networks . . 84
Example: Configuring Source Dynamic and Destination Static Translation 86
Example: Configuring NAT-PT 86
Example: Configuring an Oversubscribed Pool with Fallback to NAPT 99
Example: Configuring an Oversubscribed Pool with No Fallback 99
Example: Assigning Addresses from a Dynamic Pool for Static Use 100
Example: Configuring NAT Rules Without Defining a Pool 101
Example: Preventing Translation of Specific Addresses 101
Example: Configuring NAT for Multicast Traffic 102
Rendezvous Point Configuration 102
Router 1 Configuration 105
Example: Configuring Twice NAT 106
Example: Configuring Port Forwarding with Twice NAT 106
Example: NAT 44 CGN Configurations 108
Example: NAT Between VRFs Configuration 111
Example: Configuring Stateful NAT64 for Handling IPv4 Address Depletion 114

Chapter 6 NAT Configuration Statements 125

address 125
address-allocation 125
address-range 126
allow-overlapping-nat-pools 126
application-sets 127
applications 127
cgn-pic 128
destination-address 128
destination-address-range 129
destination-pool 129
destination-port range 130
destination-prefix 130
destination-prefix-list 131
destined-port 131
deterministic-port-block-allocation 132

dns-alg-pool . . . 132
dns-alg-prefix . . . 133
from . . . 133
hint . . . 134
ipv6-multicast-interfaces . . . 135
match-direction . . . 135
nat-type . . . 136
no-translation . . . 136
overload-pool . . . 137
overload-prefix . . . 137
pgcp . . . 138
pool . . . 139
port . . . 140
port-forwarding . . . 141
port-forwarding-mappings . . . 141
ports-per-session . . . 142
remotely-controlled . . . 142
rule . . . 143
rule-set . . . 144
services . . . 144
secured-port-block-allocation . . . 145
source-address . . . 146
source-address-range . . . 146
source-pool . . . 147
source-prefix . . . 147
source-prefix-list . . . 148
syslog . . . 148
translated-port . . . 149
term . . . 150
then . . . 151
translated . . . 152
translation-type . . . 153
translation-type (Twice NAT) . . . 155
transport . . . 156
use-dns-map-for-destination-translation . . . 156

Chapter 7 Softwire Configuration Tasks . . . 157

Configuring a DS-Lite Softwire Concentrator . . . 157
Configuring a 6rd Softwire Concentrator . . . 158
Configuring Stateful Firewall Rules for 6rd Softwire . . . 158
Configuring Softwire Rules . . . 159
Configuring Service Sets for Softwire . . . 159

Chapter 8 Softwire Configuration Examples . . . 161

Example: Basic DS-Lite Configuration . . . 161
Example: Basic 6rd Configuration . . . 166
Example: Configuring DS-Lite and 6rd in the Same Service Set . . . 169

Chapter 9 6to4 Configuration . . . 177

Configuring a 6to4 Provider-Managed Tunnel . . . 177

Chapter 10 Softwire Configuration Statements . . . 181
ds-lite . . . 182
rule (Softwire) . . . 183
rule-set (Softwire) . . . 183
softwire-concentrator . . . 184
softwire-rules . . . 184
v6rd . . . 185

Part 3 Administration

Chapter 11 Monitoring CGN and Softwire Tunnels . . . 189
Monitoring CGN, Stateful Firewall, and Softwire Flows . . . 189
Monitoring Stateful Firewall Conversations . . . 190
Monitoring Global Stateful Firewall Statistics . . . 190
Monitoring NAT Pool Usage . . . 190
Monitoring Softwire Statistics . . . 191
Ping and Traceroute for DS-Lite . . . 192

Chapter 12 Logging . . . 193
Log Generation . . . 193

Chapter 13 High-Availability and Load Balancing . . . 195
High Availability for Softwires Using Services PIC Redundancy . . . 195
Load Balancing a 6rd Domain Across Multiple Services PICs . . . 195
Example: Load Balancing a 6rd Domain Across Multiple Services PICs . . . 195
Configuring High Availability for 6rd Using 6rd Anycast . . . 200

Chapter 14 Network Address Translation Operational Mode Commands . . . 201
clear services inline nat pool . . . 202
clear services inline nat statistics . . . 203
show services inline nat pool . . . 204
show services inline nat statistics . . . 205
show services nat ipv6-multicast-interfaces . . . 206
show services nat pool . . . 208
show services nat mapping . . . 211
show services softwire . . . 213
show services softwire flows . . . 214
show services softwire statistics . . . 217
show services stateful-firewall conversations . . . 221
show services stateful-firewall flows . . . 225
show services stateful-firewall statistics . . . 230

Part 4 Index

Index . . . 237

List of Figures

Part 1 Overview

Chapter 1 Next-Generation Network Addressing 3

Figure 1: IPv4 Depletion Solution - IPv4 Access Network 4
Figure 2: IPv4 Depletion Solution - IPv6 Access Network 4

Chapter 2 Carrier-Grade NAT Solutions 7

Figure 3: Dynamic NAT Flow 11
Figure 4: Stateful NAT64 Flow 11
Figure 5: DS-Lite Flow 12

Chapter 3 Tunneling Solutions 13

Figure 6: 6rd Softwire Flow 16

Part 2 Configuration

Chapter 5 NAT Rules Examples 75

Figure 7: Configuring DNS ALGs with NAT-PT Network Topology 87
Figure 8: Configuring NAT for Multicast Traffic 102
Figure 9: NAT64 Topology 115

Chapter 8 Softwire Configuration Examples 161

Figure 10: DS-Lite Topology 162

List of Tables

About the Documentation xiii

Table 1: Notice Icons xv
Table 2: Text and Syntax Conventions xv

Part 2 Configuration

Chapter 4 NAT Configuration Tasks 21

Table 3: Comparison of NAPT Implementation Methods 29

Part 3 Administration

Chapter 14 Network Address Translation Operational Mode Commands 201

Table 4: show services inline nat pool Output Fields 204
Table 5: show services inline nat statistics Output Fields 205
Table 6: show services nat ipv6-multicast-interfaces Output Fields 206
Table 7: show services nat pool Output Fields 208
Table 8: show services nat mapping Output Fields 211
Table 9: show-services-softwire Output Fields 213
Table 10: show services softwire flows Output Fields 214
Table 11: command-name Output Fields 217
Table 12: show services stateful-firewall conversations Output Fields 223
Table 13: show services stateful-firewall flows Output Fields 227
Table 14: show services stateful-firewall statistics Output Fields 230

About the Documentation

- Documentation and Release Notes on page xiii
- Supported Platforms on page xiii
- Using the Examples in This Manual on page xiii
- Documentation Conventions on page xv
- Documentation Feedback on page xvii
- Requesting Technical Support on page xvii

Documentation and Release Notes

To obtain the most current version of all Juniper Networks® technical documentation, see the product documentation page on the Juniper Networks website at http://www.juniper.net/techpubs/.

If the information in the latest release notes differs from the information in the documentation, follow the product Release Notes.

Juniper Networks Books publishes books by Juniper Networks engineers and subject matter experts. These books go beyond the technical documentation to explore the nuances of network architecture, deployment, and administration. The current list can be viewed at http://www.juniper.net/books .

Supported Platforms

For the features described in this document, the following platforms are supported:

- M Series
- T Series
- MX Series
- J Series

Using the Examples in This Manual

If you want to use the examples in this manual, you can use the **load merge** or the **load merge relative** command. These commands cause the software to merge the incoming configuration into the current candidate configuration. The example does not become active until you commit the candidate configuration.

If the example configuration contains the top level of the hierarchy (or multiple hierarchies), the example is a *full example*. In this case, use the **load merge** command.

If the example configuration does not start at the top level of the hierarchy, the example is a *snippet*. In this case, use the **load merge relative** command. These procedures are described in the following sections.

Merging a Full Example

To merge a full example, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration example into a text file, save the file with a name, and copy the file to a directory on your routing platform.

   For example, copy the following configuration to a file and name the file **ex-script.conf**. Copy the **ex-script.conf** file to the **/var/tmp** directory on your routing platform.

   ```
   system {
     scripts {
       commit {
         file ex-script.xsl;
       }
     }
   }
   interfaces {
     fxp0 {
       disable;
       unit 0 {
         family inet {
           address 10.0.0.1/24;
         }
       }
     }
   }
   ```

2. Merge the contents of the file into your routing platform configuration by issuing the **load merge** configuration mode command:

   ```
   [edit]
   user@host# load merge /var/tmp/ex-script.conf
   load complete
   ```

Merging a Snippet

To merge a snippet, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration snippet into a text file, save the file with a name, and copy the file to a directory on your routing platform.

   For example, copy the following snippet to a file and name the file **ex-script-snippet.conf**. Copy the **ex-script-snippet.conf** file to the **/var/tmp** directory on your routing platform.

   ```
   commit {
     file ex-script-snippet.xsl; }
   ```

2. Move to the hierarchy level that is relevant for this snippet by issuing the following configuration mode command:

   ```
   [edit]
   user@host# edit system scripts
   [edit system scripts]
   ```

3. Merge the contents of the file into your routing platform configuration by issuing the **load merge relative** configuration mode command:

   ```
   [edit system scripts]
   user@host# load merge relative /var/tmp/ex-script-snippet.conf
   load complete
   ```

For more information about the **load** command, see the *Junos OS CLI User Guide*.

Documentation Conventions

Table 1 on page xv defines notice icons used in this guide.

Table 1: Notice Icons

Icon	Meaning	Description
	Informational note	Indicates important features or instructions.
	Caution	Indicates a situation that might result in loss of data or hardware damage.
	Warning	Alerts you to the risk of personal injury or death.
	Laser warning	Alerts you to the risk of personal injury from a laser.

Table 2 on page xv defines the text and syntax conventions used in this guide.

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents text that you type.	To enter configuration mode, type the **configure** command: user@host> **configure**
`Fixed-width text like this`	Represents output that appears on the terminal screen.	`user@host>` **show chassis alarms** `No alarms currently active`

Table 2: Text and Syntax Conventions *(continued)*

Convention	Description	Examples
Italic text like this	• Introduces important new terms. • Identifies book names. • Identifies RFC and Internet draft titles.	• A policy *term* is a named structure that defines match conditions and actions. • *Junos OS System Basics Configuration Guide* • RFC 1997, *BGP Communities Attribute*
Italic text like this	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name: [edit] root@# **set system domain-name** *domain-name*
Text like this	Represents names of configuration statements, commands, files, and directories; interface names; configuration hierarchy levels; or labels on routing platform components.	• To configure a stub area, include the **stub** statement at the **[edit protocols ospf area area-id]** hierarchy level. • The console port is labeled **CONSOLE**.
< > (angle brackets)	Enclose optional keywords or variables.	**stub** <**default-metric** *metric*>**;**
| (pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	**broadcast | multicast** (***string1*** **|** ***string2*** **|** ***string3***)
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	**rsvp { # Required for dynamic MPLS only**
[ ] (square brackets)	Enclose a variable for which you can substitute one or more values.	**community name members [** ***community-ids*** **]**

Indention and braces ({ })	Identify a level in the configuration hierarchy.	[edit]
		routing-options {
		  static {
		  route default {
		  nexthop *address*;
		  retain;
		  }
		  }
		}

; (semicolon)	Identifies a leaf statement at a configuration hierarchy level.	
J-Web GUI Conventions		
Bold text like this	Represents J-Web graphical user interface (GUI) items you click or select.	• In the Logical Interfaces box, select **All Interfaces**. • To cancel the configuration, click **Cancel**.
> (bold right angle bracket)	Separates levels in a hierarchy of J-Web selections.	In the configuration editor hierarchy, select **Protocols>Ospf**.

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation. You can send your comments to techpubs-comments@juniper.net, or fill out the documentation feedback form at https://www.juniper.net/cgi-bin/docbugreport/ . If you are using e-mail, be sure to include the following information with your comments:

- Document or topic name
- URL or page number
- Software release version (if applicable)

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or JNASC support contract, or are covered under warranty, and need post-sales technical support, you can access our tools and resources online or open a case with JTAC.

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the *JTAC User Guide* located at http://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf .
- Product warranties—For product warranty information, visit http://www.juniper.net/support/warranty/ .
- JTAC hours of operation—The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

For quick and easy problem resolution, Juniper Networks has designed an online self-service portal called the Customer Support Center (CSC) that provides you with the following features:

- Find CSC offerings: http://www.juniper.net/customers/support/
- Search for known bugs: http://www2.juniper.net/kb/
- Find product documentation: http://www.juniper.net/techpubs/
- Find solutions and answer questions using our Knowledge Base: http://kb.juniper.net/
- Download the latest versions of software and review release notes: http://www.juniper.net/customers/csc/software/
- Search technical bulletins for relevant hardware and software notifications: https://www.juniper.net/alerts/

- Join and participate in the Juniper Networks Community Forum: http://www.juniper.net/company/communities/
- Open a case online in the CSC Case Management tool: http://www.juniper.net/cm/

To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool: https://tools.juniper.net/SerialNumberEntitlementSearch/

Opening a Case with JTAC

You can open a case with JTAC on the Web or by telephone.

- Use the Case Management tool in the CSC at http://www.juniper.net/cm/ .
- Call 1-888-314-JTAC (1-888-314-5822 toll-free in the USA, Canada, and Mexico).

For international or direct-dial options in countries without toll-free numbers, see http://www.juniper.net/support/requesting-support.html .

PART 1

Overview

- Next-Generation Network Addressing on page 3
- Carrier-Grade NAT Solutions on page 7
- Tunneling Solutions on page 13

CHAPTER 1

Next-Generation Network Addressing

- Next-Generation Network Addressing Overview on page 3
- Sample IPv6 Transition Scenarios on page 3

Next-Generation Network Addressing Overview

In early 2011, the Internet Assigned Numbers Authority (IANA) allocated the last large block of IPv4 addresses. Now service providers and large enterprises, as well as cloud providers, e-tailers, and federal agencies, are evaluating technologies to help them avoid IPv4 address exhaustion and ensure uninterrupted subscriber and service growth.

Next-Generation Network Addressing is Juniper Networks' portfolio of IPv4 exhaustion avoidance, IPv4-IPv6 coexistence, and IPv6 transition technologies that include IPv6, v4/v6 dual stack, NAT44, NAT44(4), NAPT44, NAPT444, NAT-PT, NAT64, 6-to4-PMT, 6rd, and DS-Lite. These technologies help network operators improve subscriber and service scale, mitigate IPv4 address depletion, and pragmatically transition to IPv6 based on business requirements.

Sample IPv6 Transition Scenarios

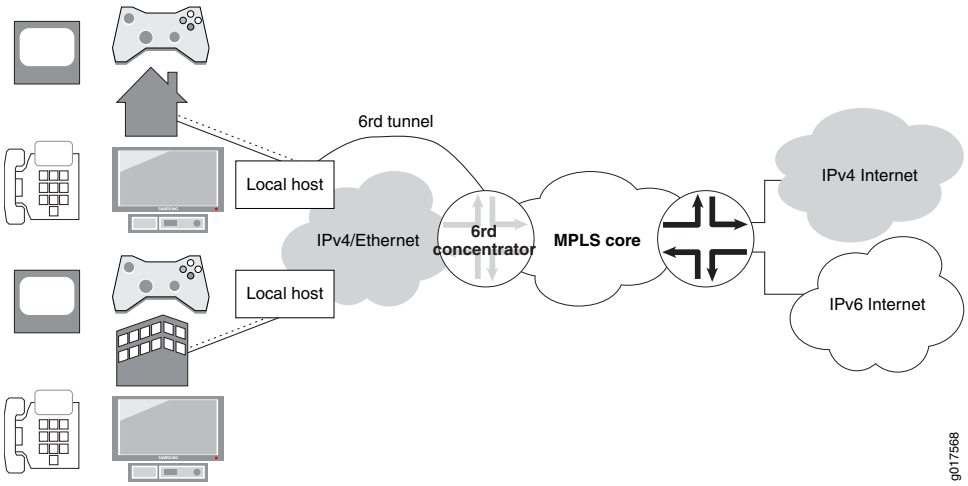
The Junos OS supports many IPv6 transition scenarios required by Junos OS customers. The following are selected examples:

- Example 1: IPv4 Depletion with a Non-IPv6 Access Network on page 3
- Example 2: IPv4 Depletion with an IPv6 Access Network on page 4
- Example 3: IPv4 Depletion for Mobile Networks on page 5

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

Figure 1 on page 4 depicts a scenario in which the Internet service provider (ISP) has not significantly changed its IPv4 network. This approach enables IPv4 hosts to access the IPv4 Internet and IPv6 hosts to access the IPv6 Internet. A dual-stack host can be treated as an IPv4 host when it uses the IPv4 access service, and as an IPv6 host when it uses the IPv6 access service.

Figure 1: IPv4 Depletion Solution - IPv4 Access Network

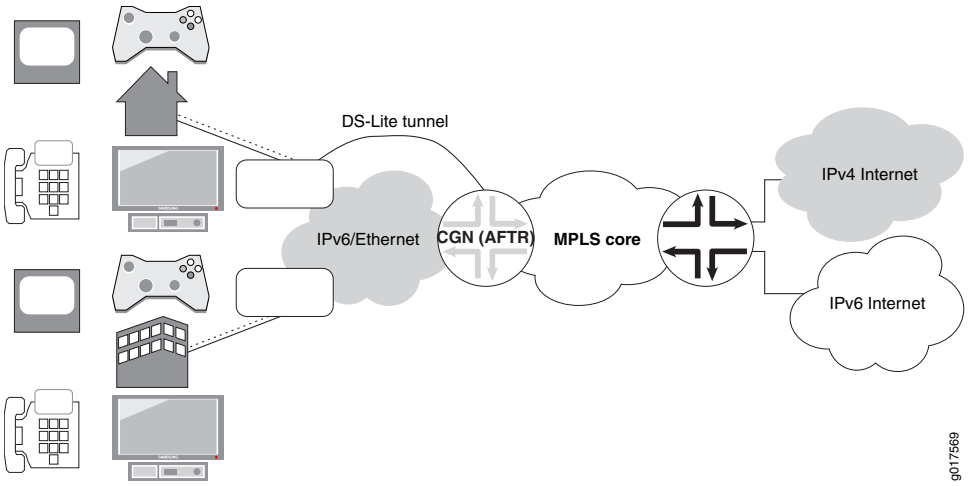


Two new types of devices must be deployed in this approach: a dual-stack home gateway and a dual-stack carrier-grade Network Address Translation (NAT). The dual-stack home gateway integrates IPv4 forwarding and v6-over-v4 tunneling functions. It can also integrate a v4-v4 NAT function. The dual-stack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

Example 2: IPv4 Depletion with an IPv6 Access Network

In the scenario shown in Figure 2 on page 4, the ISP network is IPv6-only.

Figure 2: IPv4 Depletion Solution - IPv6 Access Network



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

Not all customers of a given ISP must switch from IPv4 access to IPv6 access simultaneously; in fact, transition can be managed better by switching groups of

customers (for example, all those connected to a single point of presence) on an incremental basis. Such an incremental approach should prove easier to plan, schedule, and execute than an across-the-board cut-over.

Example 3: IPv4 Depletion for Mobile Networks

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

CHAPTER 2

Carrier-Grade NAT Solutions

- Network Address Translation Overview on page 7

Network Address Translation Overview

- Types of NAT on page 7

Types of NAT

The types of Network Address Translation supported by the Junos OS are described in the following sections:

- NAT Concept and Facilities Overview on page 7
- IPv4-to-IPv4 Basic NAT on page 8
- NAT-PT on page 9
- Static Destination NAT on page 9
- Twice NAT on page 9
- IPv6 NAT on page 10
- NAT-PT with DNS ALG on page 10
- Dynamic NAT on page 10
- Stateful NAT64 on page 11
- Dual-Stack Lite on page 11

NAT Concept and Facilities Overview

NAT is a mechanism for translating IP addresses. NAT provides the technology used to support a wide range of networking goals, including:

- Concealing a set of host addresses on a private network behind a pool of public addresses.
- Providing a security measure to protect the host addresses from direct targeting in network attacks.
- Providing a tool set for coping with IPv4 address depletion and IPV6 transition issues.

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

The multiservices Dense Port Concentrator (DPC) and multiservices PIC interfaces support the following types of traditional CGN:

- Static-source translation—Allows you to hide a private network. It features a one-to-one mapping between the original address and the translated address; the mapping is configured statically. For more information, see "Basic NAT" on page 8.
- Dynamic-source translation—Includes two options: dynamic address-only source translation and Network address Port Translation (NAPT):
  - Dynamic address-only source translation—A NAT address is picked up dynamically from a source NAT pool and the mapping from the original source address to the translated address is maintained as long as there is at least one active flow that uses this mapping. For more information, see "Dynamic NAT" on page 10.
  - NAPT—Both the original source address and the source port are translated. The translated address and port are picked up from the corresponding NAT pool. For more information, see "NAPT" on page 9.
- Static destination translation—Allows you to make selected private servers accessible. It features a one-to-one mapping between the translated address and the destination address; the mapping is configured statically. For more information, see "Static Destination NAT" on page 9.
- Protocol translation—Allows you to assign addresses from a pool on a static or dynamic basis as sessions are initiated across IPv4 or IPv6 boundaries. For more information, see "NAT-PT" on page 9, "NAT-PT with DNS ALG" on page 10, and "Stateful NAT64" on page 11.
- Encapsulation of IPv4 packets into IPv6 packets using softwires—Enables packets to travel over softwires to a carrier-grade NAT endpoint where they undergo source-NAT processing to hide the original source address. For more information, see "Tunneling Services for IPv4-to-IPv6 Transition Overview" on page 13.

The Junos OS supports NAT functionality described in IETF RFCs and Internet drafts, as shown in "Supported NAT and SIP Standards" in *Standards Supported in Junos OS 11.4*.

IPv4-to-IPv4 Basic NAT

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

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

Basic NAT

With Basic NAT, a block of external addresses are set aside for translating addresses of hosts in a private domain as they originate sessions to the external domain. For packets

outbound from the private network, Basic NAT translates source IP addresses and related fields such as IP, TCP, UDP, and ICMP header checksums. For inbound packets, Basic NAT translates the destination IP address and the checksums listed above.

NAPT

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

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

NAT-PT

NAT-Protocol Translation (NAT-PT) is an obsolete IPv4-to-IPv6 transition mechanism and is no longer recommended. NAT64 is the newer, recommended solution. Using a pool of IPv4 addresses, NAT-PT assigns addresses from that pool to IPv6 nodes on a dynamic basis as sessions are initiated across IPv4 or IPv6 boundaries. Inbound and outbound sessions must traverse the same NAT-PT router so that it can track those sessions. RFC 2766, *Network Address Translation - Protocol Translation (NAT-PT)*, recommends the use of NAT-PT for translation between IPv6-only nodes and IPv4-only nodes, and *not* for IPv6-to-IPv6 translation between IPv6 nodes or IPv4-to-IPv4 translation between IPv4 nodes.

NAT-PT, specified in RFC 2766, *Network Address Translation - Protocol Translation (NAT-PT)* and obsoleted by RFC 2766, *Reasons to Move Network Address Translator - Protocol Translator (NAT-PT) to Historic Status*, is still supported by the the Junos OS.

Static Destination NAT

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

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

Twice NAT

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

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

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

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

IPv6 NAT

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

NAT-PT with DNS ALG

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

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

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

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

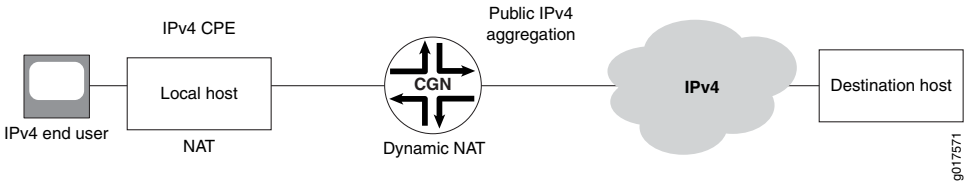
Related Documentation

- Configuring NAT Rules on page 30
- Configuring NAT-PT on page 58
- Example: Configuring NAT-PT on page 86

Dynamic NAT

Dynamic NAT flow is shown in Figure 3 on page 11.

Figure 3: Dynamic NAT Flow



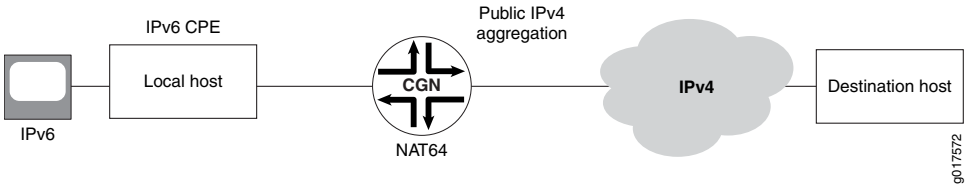
With dynamic NAT, you can map a private IP address (source) to a public IP address drawing from a pool of registered (public) IP addresses. NAT addresses from the pool are assigned dynamically. Assigning addresses dynamically also allows a few public IP addresses to be used by several private hosts,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

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

Stateful NAT64

Stateful NAT64 flow is shown in Figure 4 on page 11.

Figure 4: Stateful NAT64 Flow



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

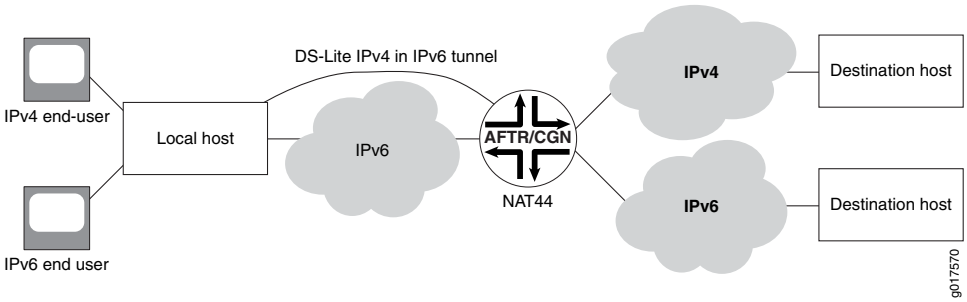
When stateful NAT64 is used in conjunction with DNS64, no changes are usually required in the IPv6 client or the IPv4 server. DNS64 is out of scope of this document because it is normally implemented as an enhancement to currently deployed DNS servers.

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

Dual-Stack Lite

Dual-stack lite (DS-Lite) flow is shown in Figure 5 on page 12.

Figure 5: DS-Lite Flow



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

Related Documentation

- DS-Lite Softwires—IPv4 over IPv6
- Configuring a DS-Lite Softwire Concentrator on page 157

CHAPTER 3

Tunneling Solutions

- Tunneling Services for IPv4-to-IPv6 Transition Overview on page 13

Tunneling Services for IPv4-to-IPv6 Transition Overview

The Junos OS enables service providers to transition to IPv6 by using softwire encapsulation and decapsulation techniques. A softwire is a tunnel that is created between softwire Customer Premises Equipment (CPE). A softwire CPE can share a unique common internal state for multiple softwires, making it a very light and scalable solution. When you use softwires, you need not maintain an interface infrastructure for each softwire, unlike a typical mesh of generic routing encapsulation (GRE) tunnels that would require you to do so. A softwire initiator at the customer end encapsulates native packets and tunnels them to a softwire concentrator at the service provider. The softwire concentrator decapsulates the packets and sends them to their destination. A softwire is created when a softwire concentrator receives the first tunneled packet of a flow and prepares for flow processing. The softwire exists as long as the softwire concentrator is providing flows for routing. A flow counter is maintained; when the number of active flows is 0, the softwire is deleted. Statistics are kept for both flows and softwires.

Softwire addresses are not specifically configured under any physical or virtual interface. Therefore, the number of established softwires does not affect throughput, and scalability is independent of the number of interfaces. The scalability is only limited to the number of flows that the platform (services DPC or PIC) can support.

This topic contains the following sections:

- 6to4 Overview on page 13
- DS-Lite Softwires—IPv4 over IPv6 on page 15
- 6rd Softwires—IPv6 over IPv4 on page 16

6to4 Overview

- Basic 6to4 on page 14
- 6to4 Anycast on page 14
- 6to4 Provider-Managed Tunnels on page 15

Basic 6to4

6to4 is an Internet transition mechanism for migrating from IPv4 to IPv6, a system that allows IPv6 packets to be transmitted over an IPv4 network (generally the IPv4 Internet) without the need to configure explicit tunnels. 6to4 is described in RFC 3056, *Connection of IPv6 Domains via IPv4 Clouds*. 6to4 is especially relevant during the initial phases of deployment to full, native IPv6 connectivity, since IPv6 is not required on nodes between the host and the destination. However, it is intended only as a transition mechanism and is not meant to be used permanently.

6to4 can be used by an individual host, or by a local IPv6 network. When used by a host, it must have a global IPv4 address connected, and the host is responsible for the encapsulation of outgoing IPv6 packets and the decapsulation of incoming 6to4 packets. If the host is configured to forward packets for other clients, often a local network, it is then a router.

There are two kinds of 6to4 virtual routers: border routers and relay routers. A 6to4 border router is an IPv6 router supporting a 6to4 pseudointerface, and is normally the border router between an IPv6 site and a wide-area IPv4 network. A relay router is a 6to4 router configured to support transit routing between 6to4 addresses and pure native IPv6 addresses.

In order for a 6to4 host to communicate with the native IPv6 Internet, its IPv6 default gateway must be set to a 6to4 address which contains the IPv4 address of a 6to4 relay router. To avoid the need for users to set this up manually, the Anycast address of 192.88.99.1 has been allocated to send packets to a 6to4 relay router. Note that when wrapped in 6to4 with the subnet and hosts fields set to zero, this IPv4 address (192.88.99.1) becomes the IPv6 address 2002:c058:6301::. To ensure BGP routing propagation, a short prefix of 192.88.99.0/24 has been allocated for routes pointed at 6to4 relay routers that use this Anycast IP address. Providers willing to provide 6to4 service to their clients or peers should advertise the Anycast prefix like any other IP prefix, and route the prefix to their 6to4 relay.

Packets from the IPv6 Internet to 6to4 systems must be sent to a 6to4 relay router by normal IPv6 routing methods. The specification states that such relay routers must only advertise 2002::/16 and not subdivisions of it to prevent IPv4 routes from polluting the routing tables of IPv6 routers. From there they can then be sent over the IPv4 Internet to the destination.

6to4 Anycast

Router 6to4 assumes that 6to4 routers and relays are managed and configured cooperatively. In particular, 6to4 sites must configure a relay router to carry the outbound traffic, which becomes the default IPv6 router (except for 2002::/16). The objective of the Anycast variant, defined in RFC 3068, *An Anycast Prefix for 6to4 Relay Routers*, is to avoid the need for such configuration. This makes the solution available for small or domestic users, even those with a single host or simple home gateway instead of a border router. This is achieved by defining 192.88.99.1 as the default IPv4 address for a 6to4 relay, and 2002:c058:6301:: as the default IPv6 router prefix (“well-known prefix”) for a 6to4 site.

RFC 6343, *Advisory Guidelines for 6to4 Deployment*, published in August 2011, identifies a wide range of problems associated with the use of unmanaged 6to4 Anycast relay routers.

6to4 Provider-Managed Tunnels

A solution to many problems associated with unmanaged Anycast 6to4 is presented in IETF informational draft draft-kuarsingh-v6ops-6to4-provider-managed-tunnel-02, *6to 4 Provider-Managed Tunnels (PMT)*. That document, a "work in progress," proposes a solution that allows providers to exercise greater control over the routing of 6to4 traffic.

Anycast 6to4 implies a default configuration for the user site. It does not require any particular user action. It does require an IPv4 Anycast route to be in place to a relay at 192.88.99.1. Traffic does not necessarily return to the same 6to4 gateway because of the the "well-known" 6to4 prefix used and advertised by all 6to4 traffic.

6to4 provider-managed tunnels (PMTs) facilitate the management of 6to4 tunnels using an Anycast configuration. 6to4 PMT enables service providers to improve 6to4 operation when network conditions provide suboptimal performance or break normal 6to4 operation. 6to4 PMT provides a stable provider prefix and forwarding environment by utilizing existing 6to4 relays with an added function of IPv6 prefix translation that controls the flow of return traffic.

The 6to4 managed tunnel model behaves like a standard 6to4 service between the customer IPv6 host or gateway and the 6to4 PMT relay (within the provider domain). The 6to4-PMT relay shares properties with 6rd (RFC5969) by decapsulating and forwarding embedded IPv6 flows, within an IPv4 packet, to the IPv6 Internet. The model provides an additional function which translates the source 6to4 prefix to a provider assigned prefix which is not found in 6rd (RFC5969) or traditional 6to4 operation. The 6to4-PMT relay provides a stateless (or stateful) mapping of the 6to4 prefix to a provider-supplied prefix by mapping the embedded IPv4 address in the 6to4 prefix to the provider prefix.

DS-Lite Softwires—IPv4 over IPv6

When an Internet service provider (ISP) begins to allocate new subscriber home IPv6 addresses and IPv6-capable equipment, dual-stack lite (DS-Lite) provides a method for the private IPv4 addresses behind the IPv6 customer edge (CE) WAN equipment to reach the IPv4 network. DS-Lite enables IPv4 customers to continue to access the Internet using their current hardware by using a softwire initiator, referred to as a Basic Bridging Broadband (B4), at the customer edge to encapsulate IPv4 packets into IPv6 packets and tunnel them over an IPv6 network to a softwire concentrator, referred to as an Address Family Transition Router (AFTR), for decapsulation. DS-Lite creates the IPv6 softwires that terminate on the services PIC. Packets coming out of the softwire can then have other services such as NAT applied on them.

DS-Lite is supported on Multiservices 100, 400, and 500 PICs on M Series routers and on MX Series routers equipped with Multiservices Dense Port Concentrator (DPCs).

NOTE: IPv6 Provider Edge (6PE), or MPLS-enabled IPv6, is available for ISPs with MPLS-enabled networks. These networks now can use Multiprotocol Border Gateway Protocol (MP-BGP) to provide connectivity between the DS-Lite B4 and AFTR (or any two IPv6 nodes). DS-Lite properly handles encapsulation and decapsulation despite the presence of additional MPLS header information.

For more information on DS-Lite softwires, see the IETF draft *Dual Stack Lite Broadband Deployments Following IPv4 Exhaustion*.

NOTE: The most recent IETF draft documentation for DS-Lite uses new terminology:

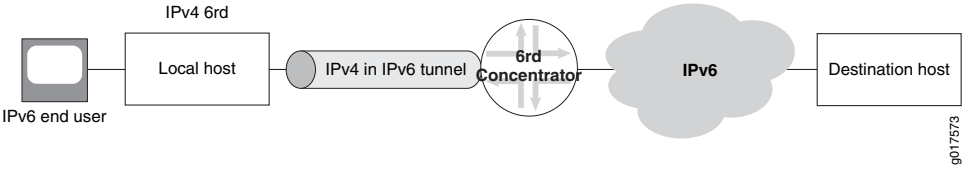
- The term *softwire initiator* has been replaced by *B4*.
- The term *softwire concentrator* has been replaced by *AFTR*.

The Junos OS documentation generally uses the original terms when discussing configuration in order to be consistent with the command-line interface (CLI) statements used to configure DS-Lite.

6rd Softwires—IPv6 over IPv4

6rd softwire flow is shown in Figure 6 on page 16.

Figure 6: 6rd Softwire Flow



The Junos OS supports a 6rd softwire concentrator on a services DPC or PIC to facilitate rapid deployment of IPv6 service to subscribers on native IPv4 CE WANs. IPv6 packets are encapsulated in IPv4 packets by a softwire initiator at the CE WAN. These packets are tunneled to a softwire concentrator residing on a multiservices DPC (branch relay). A softwire is created when IPv4 packets containing IPv6 destination information are received at the softwire concentrator, which decapsulates IPv6 packets and forwards them for IPv6 routing. All of these functions are performed in a single pass of the services PIC.

In the reverse path, IPv6 packets are sent to the Services DPC where they are encapsulated in IPv4 packets corresponding to the proper softwire and sent to the CE WAN.

The softwire concentrator creates softwires as the IPv4 packets are received from the CE WAN side or IPV6 packets are received from the Internet. A 6rd softwire on the Services DPC is identified by the 3-tuple containing the service set ID, CE softwire initiator IPv4 address, and softwire concentrator IPv4 address. IPv6 flows are also created for the

encapsulated IPv6 payload, and are associated with the specific softwire that carried them in the first place. When the last IPv6 flow associated with a softwire ends, the softwire is deleted. This simplifies configuration and there is no need to create or manage tunnel interfaces.

6rd is supported on Multiservices 100, 400, and 500 PICs on M Series and T Series routers, and on MX Series platforms equipped with Multiservices DPCs.

For more information on 6rd softwires, see RFC 5969, *IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) -- Protocol Specification*.

Related Documentation

- See Network Address Translation Overview on page 7.

PART 2

Configuration

- NAT Configuration Tasks on page 21
- NAT Rules Examples on page 75
- NAT Configuration Statements on page 125
- Softwire Configuration Tasks on page 157
- Softwire Configuration Examples on page 161
- 6to4 Configuration on page 177
- Softwire Configuration Statements on page 181

CHAPTER 4

NAT Configuration Tasks

- Configuring Addresses and Ports for Use in NAT Rules on page 21
- Configuring NAT Rules on page 30
- Configuring NAT Rule Sets on page 35
- Configuring NAT Service Sets on page 36
- Configuring Static Source Translation in IPv4 Networks on page 37
- Configuring Static Source Translation in IPv6 Networks on page 41
- Configuring Dynamic Source Address and Port Translation in IPv4 Networks on page 44
- Configuring Dynamic Source Address and Port Translation for IPv6 Networks on page 47
- Configuring Dynamic Address-Only Source Translation in IPv4 Networks on page 48
- Configuring Static Destination Address Translation in IPv4 Networks on page 51
- Configuring Translation Type for Translation Between IPv6 and IPv4 Networks on page 53
- Configuring NAT-PT on page 58
- Example: Configuring Port Forwarding with Twice NAT on page 60
- Configuring NAT-PT on page 62
- Configuring Dynamic Source Address and Static Destination Address Translation (IPv6 to IPV4) on page 65
- Configuring Port Forwarding for Static Destination Address Translation on page 66
- Configuring Port Forwarding Without Destination Address Translation on page 69
- Configuring Secured Port Block Allocation on page 70
- Configuring Deterministic Port Block Allocation on page 72

Configuring Addresses and Ports for Use in NAT Rules

For information about configuring translated addresses, see the following sections:

- Configuring Pools of Addresses and Ports on page 22
- Configuring Address Pools for Network Address Port Translation on page 23
- Specifying Destination and Source Prefixes on page 29
- Requirements for NAT Addresses on page 29

Configuring Pools of Addresses and Ports

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

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

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

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

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

For constraints on specific translation types, see "Configuring Actions in NAT Rules" on page 33.

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

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

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

Preserve Range and Preserve Parity

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

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

Configuring Address Pools for Network Address Port Translation

With Network Address Port Translation (NAPT), you can configure up to 32 address ranges with up to 65,536 addresses each.

The **port** statement specifies port assignment for the translated addresses. To configure automatic assignment of ports, include the **port automatic** statement at the **[edit services nat pool *nat-pool-name*]** hierarchy level. To configure a specific range of port numbers, include the **port range low *minimum-value* high *maximum-value*** statement at the **[edit services nat pool *nat-pool-name*]** hierarchy level.

The Junos OS provides several alternatives for allocating ports:

- Round-Robin Allocation on page 23
- Sequential on page 24
- Port Block Allocation on page 25
- Additional Options for NAPT on page 28
- Comparision of NAPT Implementation Methods on page 28

Round-Robin Allocation

To configure round-robin allocation for NAT pools, include the **address-allocation round-robin** configuration statement at the **[edit services nat pool *pool-name*]** hierarchy level. When you use round-robin allocation, one port is allocated from each address in a range before repeating the process for each address in the next range. After ports have been allocated for all addresses in the last range, the allocation process wraps around and allocates the next unused port for addresses in the first range.

- The first connection is allocated to the address:port 100.0.0.1:3333.
- The second connection is allocated to the address:port 100.0.0.2:3333.
- The third connection is allocated to the address:port 100.0.0.3:3333.
- The fourth connection is allocated to the address:port 100.0.0.4:3333.

- The fifth connection is allocated to the address:port 100.0.0.5:3333.
- The sixth connection is allocated to the address:port 100.0.0.6:3333.
- The seventh connection is allocated to the address:port 100.0.0.7:3333.
- The eighth connection is allocated to the address:port 100.0.0.8:3333.
- The ninth connection is allocated to the address:port 100.0.0.9:3333.
- The tenth connection is allocated to the address:port 100.0.0.10:3333.
- The eleventh connection is allocated to the address:port 100.0.0.11:3333.
- The twelfth connection is allocated to the address:port 100.0.0.12:3333.
- Wraparound occurs and the thirteenth connection is allocated to the address:port 100.0.0.1:3334.

Sequential

With sequential allocation, the next available address in the NAT pool is selected only when all the ports available from an address are exhausted.

NOTE: This legacy implementation provides backward compatibility.

The NAT pool called **napt** in the following configuration example uses the sequential implementation:

```
pool napt {
    address-range low 100.0.0.1 high 100.0.0.3;
    address-range low 100.0.0.4 high 100.0.0.6;
    address-range low 100.0.0.8 high 100.0.0.10;
    address-range low 100.0.0.12 high 100.0.0.13;
    port {
        range low 3333 high 3334;
    }
}
```

In this example, the ports are allocated starting from the first address in the first address-range, and allocation continues from this address until all available ports have been used. When all available ports have been used, the next address (in the same address-range or in the following address-range) is allocated and all its ports are selected as needed. In the case of the example **napt** pool, the tuple address, port 100.0.0.4:3333, is allocated only when all ports for all the addresses in the first range have been used.

- The first connection is allocated to the address:port 100.0.0.1:3333.
- The second connection is allocated to the address:port 100.0.0.1:3334.
- The third connection is allocated to the address:port 100.0.0.2:3333.
- The fourth connection is allocated to the address:port 100.0.0.2:3334, and so on.

Port Block Allocation

Carriers track subscribers using the IP address (RADIUS or DHCP) log. If they use CGN, an IP address is shared by multiple subscribers, and the carrier must track the IP address and port, which are part of the NAT log. Because ports are used and reused at a very high rate, tracking subscribers using the log becomes difficult due to the large number of messages, which are difficult to archive and correlate. By enabling the allocation of ports in blocks, port block allocation can significantly reduce the number of logs, making it easier to track subscribers.

The Junos OS provides the following methods of block allocation for ports:

- Secured Port Block Allocation on page 25
- Deterministic Port Block Allocation on page 25

Secured Port Block Allocation

When allocating blocks of ports, the most recently allocated block is the current active block. New requests for NAT ports are served from the active block. Ports are allocated randomly from the current active block.

When you configure secured port block allocation, you can specify the following:

- **block-size**
- **max-blocks-per-address**
- **active-block-timeout**

Related Documentation

- Configuring Secured Port Block Allocation on page 70

Deterministic Port Block Allocation

You can configure NAT algorithm-based allocation of blocks of destination ports. By specifying **deterministic-port-block-allocation blocksize *blocksize*** at the **[edit services nat pool *poolname* port]** hierarchy level, you ensure that an incoming (source) IP address and port always map to the same destination IP address and port, thus eliminating the need for the address translation logging. When you use deterministic port block allocation, you must specify **deterministic-nat44** as the **translation-type** in your NAT rule.

Understanding Deterministic Port Block Allocation Algorithms

The effectiveness of your implementation of deterministic port block allocation depends on your analysis of your subscriber requirements. The block size you provide indicates how many ports will be made available for each incoming subscriber address in the range the **from** clause specified in the applicable NAT rule. The allocation algorithm computes an offset value to determine the outgoing port. A reverse algorithm is used to derive the originating subscriber address.

NOTE: In order to track subscribers without using logs, an ISP must use a the reverse algorithm to derive a subscriber (source) addresses from translated addresses.

Deterministic Port Block Allocation Algorithm Usage

When you have configured deterministic port block allocation, you can use the **show services nat deterministic-nat internal-host** and **show services nat deterministic-nat nat-port-block**

to show forward and reverse mapping. However, mappings will change if you reconfigure your deterministic port block allocation block size or the **from** clause for your NAT rule. In order to provide historical information on mappings, we recommend that you write scripts that can show specific mappings for prior configurations.

The following variables are used in forward calculation (private subscriber IP address to public IP address) and reverse calculation (public IP address to private subscriber IP address):

- Pr_Prefix—Any pre-NAT IPv4 subscriber address.
- Pr_Port—Any pre-NAT protocol port.
- Block_Size—Number of ports configured to be available for each Pr_Prefix.
- Base_PR_Prefix—First usable pre-NAT IPv4 subscriber address in a “from” clause match condition.
- Base_PU_Prefix—First usable post-NAT IPv4 subscriber address configured in the NAT pool.
- Pu_Port_Range_Start—1024 (ports 0 through 1023 are not used when **port automatic** is configured)
- Pr_Offset—Pr_Prefix – Base_Pr_Prefix
- PR_Port_Offset—Pr_Offset * Block_Size
- Pu_Prefix—Post-NAT address for a given Pr_Prefix
- Pu_Start_Port—Post-NAT start port for a flow from a given Pr_Prefix
- Pu_Actual_Port—Post-NAT port seen on a reverse flow
- Port_Range_Per_Pu_IP—64,512 (65,536 – 1024), constant as privileged ports are unused due to “port automatic” configuration)
- Pu_Offset—Pu_Prefix – Base_Pu_Prefix
- Pu_Port_Offset—(Pu_Offset * Port_Range_Per_Pu_IP) + (Pu_Actual_Port – Pu_Port_Start_Port)

NOTE: If **block-size** is configured as zero, the actual block size is computed by the Junos OS by dividing the number of ports available in the source-pool by the number of subscribers in the **from** clause.

Algorithm Usage–Assume the following configuration:

```
services {
  nat {
    pool src-pool {
      address-range low 32.32.32.1 high 32.32.32.254;
      port {
        automatic {
          random-allocation;
        }
        deterministic-block-allocation {
          block-size 256;
        }
      }
    }
rule det-nat {
match-direction input;
  term t1 {
    from {
      source-address {
        10.1.0.0/16;
      }
    }
    then {
      translated {
        source-pool src-pool;
        translation-type {
          deterministic-napt44;
        }
      }
    }
  }
```

Forward Translation

1. Pr_Offset = Pr_Prefix – Base_Pr_Prefix
2. Pr_Port_Offset = Pr_Offset * Block_Size
3. Pu_Prefix = Base_Public_Prefix + (Pr_Port_Offset / Port_Range_Per_IP)
4. Pu_Start_Port = Pu_Port_Range_Start + (Pr_Port_Offset % Port_Range_Per_IP)

Using the sample configuration and assuming a subscriber flow sourced from 10.1.1.250:5000;

1. Pr_Offset = 10.1.1.250 – 10.1.0.1 = 505
2. Pu_Port_Offset = 505 * 256 = 129,280

3. Pu_Prefix = 32.32.32.1 + (129,280 / 64,512) = 32.32.32.3
4. Pu_Start_Port = 1,024 + (129,280 % 64,512) = 1,280
   - 10.1.1.250 is translated to 32.32.32.3 .
   - The starting port is 1280. There are 256 ports available to the subscriber based on the configured block size. The available port range spans ports 1280 through 1535 (inclusive).
   - The specific flow 10.1.1.250:5000 randomly assigns any of the ports in its range because random allocation was specified.

Reverse Translation

1. Pu_Offset = Pu_Prefix – Base_Pu_Prefix
2. Pu_Port_Offset = (Pu_Offset * Port_Range_Per_Pu_IP) + (Pu_Actual_Port – Pu_Port_Range_Start)
3. Subscriber_IP = Base_Pr_Prefix + (Pu_Port_Offset / Block_Size)

The reverse translation is determined as follows. Assume a flow returning to 32.32.32.3:1400.

1. Pu_Offset = 32.32.32.3 – 32.32.32.1 = 2
2. Pu_Port_Offset = (2 * 64,512) + (1400 - 1024) = 129,400
3. Subscriber_IP = 10.1.0.1 + (129,400 / 256) = 10.1.1.250

NOTE: In reverse translation, only the original private IP address can be derived, and not the original port in use. This is sufficiently granular for law enforcement requirements.

Additional Options for NAPT

The following options are available for NAPT:

- Preserving parity—Use the **preserve-parity** command to allocate even ports for packets with even source ports and odd ports for packets with odd source ports.
- Preserving range—Use the **preserve-range** command to allocate ports within a range from 0 to 1023, assuming the original packet contains a source port in the reserved range. This applies to control sessions, not data sessions.

Comparision of NAPT Implementation Methods

Table 3 on page 29 provides a feature comparison of available NAPT implementation methods.

Table 3: Comparison of NAPT Implementation Methods

Feature/Function	Dynamic Port Allocation	Secured Port Block Allocation	Deterministic Port Block Allocation
Users per IP	High	Medium	Low
Security Risk	Low	Medium	Medium
Log Utilization	High	Low	None (no logs necessary)
Security Risk Reduction	Random allocation	**active-block-timeout** feature	n/a
Increasing Users per IP	n/a	Configure multiples of smaller port blocks to maximize users/ public IP	Algorithm-based port allocation

Specifying Destination and Source Prefixes

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

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

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

Requirements for NAT Addresses

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

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

  - **240.0.0.0/4** (reserved)
  - **255.255.255.255** (broadcast)

- You can specify one or more IPv4 address prefixes in the **pool** statement and in the **from** clause of the NAT rule term. This enables you to configure source translation from a private subnet to a public subnet without defining a rule term for each address in the subnet. Destination translation cannot be configured by this method. For more information, see *Examples: Configuring NAT Rules.*
- When you configure static source NAT, the **address** prefix size you configure at the **[edit services nat pool *pool-name*]** hierarchy level must be larger than the **source-address** prefix range configured at the **[edit services nat rule *rule-name* term *term-name* from]** hierarchy level. The **source-address** prefix range must also map to a single subnet or range of IPv4 or IPv6 addresses in the **pool** statement. Any pool addresses that are not used by the **source-address** prefix range are left unused. Pools cannot be shared.

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

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

Configuring NAT Rules

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

```
[edit services nat]
rule rule-name {
    match-direction (input | output);
    term term-name {
        from (Services NAT) {
            application-sets (Services NAT) set-name;
            applications (Services NAT) [ application-names ];
            destination-address (address | any-unicast) <except>;
            destination-address-range low minimum-value high maximum-value <except>;
            destination-prefix-list list-name <except>;
            source-address (address | any-unicast) <except>;
            source-address-range low minimum-value high maximum-value <except>;
            source-prefix-list list-name <except>;
        }
        then {
            no-translation;
            translated {
                address-pooling paired;
                destination-pool nat-pool-name;
```

```
            destination-prefix destination-prefix;
            dns-alg-pool dns-alg-pool;
            dns-alg-prefix dns-alg-prefix;
            filtering-type endpoint-independent;
            mapping-type endpoint-independent;
            overload-pool overload-pool-name;
            overload-prefix overload-prefix;
            source-pool nat-pool-name;
            source-prefix source-prefix;
            translation-type {
              (basic-nat-pt | basic-nat44 | basic-nat66 | dnat-44 | dynamic-nat44 | napt-44 |
                napt-66 | napt-pt | stateful-nat64 | twice-basic-nat-44 | twice-dynamic-nat-44
                | twice-napt-44);
            }
            use-dns-map-for-destination-translation;
          }
          syslog;
        }
      }
    }
```

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

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

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

The following sections explain how to configure the components of NAT rules:

- Configuring Match Direction for NAT Rules on page 31
- Configuring Match Conditions in NAT Rules on page 32
- Configuring Actions in NAT Rules on page 33

Configuring Match Direction for NAT Rules

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

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

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

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

Configuring Match Conditions in NAT Rules

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

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

To configure traditional NAT, you can use the destination address, a range of destination addresses, the source address, or a range of source addresses as a match condition, in the same way that you would configure a firewall filter; for more information, see the *Junos OS Policy Framework Configuration Guide*.

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

You can include application protocol definitions that you have configured at the **[edit applications]** hierarchy level; for more information, see "Configuring Application Protocol Properties":

- To apply one or more specific application protocol definitions, include the **applications** statement at the **[edit services nat rule *rule-name* term *term-name* from]** hierarchy level.
- To apply one or more sets of application protocol definitions that you have defined, include the **application-sets** statement at the **[edit services nat rule *rule-name* term *term-name* from]** hierarchy level.

NOTE: If you include one of the statements that specifies application protocols, the router derives port and protocol information from the corresponding configuration at the **[edit applications]** hierarchy level; you cannot specify these properties as match conditions. When matched rules include more than one ALG, the more specific ALG takes effect; for example, if the stateful firewall rule includes TCP and the NAT rule includes FTP, the NAT rule takes precedence.

You can configure ALGs for ICMP and traceroute under stateful firewall and NAT.

By default, NAT can restore IP, TCP, and UDP headers embedded in the payload of ICMP error messages. You can include the **protocol tcp** and **protocol udp** statements with the **application** statement for NAT configurations.

Configuring Actions in NAT Rules

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

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

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

The **syslog** statement enables you to record an alert in the system logging facility.

The **destination-pool**, **destination-prefix**, **source-pool**, and **source-prefix** statements specify addressing information that you define by including the **pool** statement at the **[edit services nat]** hierarchy level; for more information, see "Configuring Addresses and Ports for Use in NAT Rules" on page 21.

The **translation-type** statement specifies the type of NAT used for source or destination traffic. The options are **basic-nat-pt**, **basic-nat44**, **basic-nat66**, **dnat-44**, **dynamic-nat44**, **napt-44**, **napt-66**, **napt-pt**, and **stateful-nat64** . For more information, see "Network Address Translation Overview" on page 7.

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

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

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

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

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

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

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

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

  The **dynamic-nat44** address-only option supports translating up to 16,777,216 addresses to a smaller size pool. The requests from the source address range are assigned to the

addresses in the pool until the pool is used up, and any additional requests are rejected. A NAT address assigned to a host is used for all concurrent sessions from that host. The address is released to the pool only after all the sessions for that host expire. This feature enables the router to share a few public IP addresses between several private hosts. Because all the private hosts might not simultaneously create sessions, they can share a few public IP addresses.

- **napt-44**—This option implements dynamic translation of source IP addresses with port mapping. You must specify a name for the **source-pool** statement. The referenced pool must include a **port** configuration. If the port is configured as automatic or a port range is specified, then it implies that Network Address Port Translation (NAPT) is used.
- **napt-66**—This option implements dynamic address translation of source IP addresses with port mapping for IPv6 addresses. The configuration is similar to the **napt-44** implementation, but with IPv6 addresses.
- **napt-pt**—This option implements dynamic address and port translation for source and static translation of destination IP address. You must specify a name for the **source-pool** statement. The referenced pool must include a port configuration (for NAPT). Additionally, you must configure two rules, one for the DNS traffic and the other for the rest of the traffic. The rule meant for the DNS traffic should be DNS ALG enabled and the **dns-alg-prefix** statement should be configured. Moreover, the prefix configured in the **dns-alg-prefix** statement must be used in the second rule to translate the destination IPv6 addresses to IPv4 addresses.
- **stateful-nat64**—This option implements dynamic address and port translation for source IP addresses and prefix removal translation for destination IP addresses. You must specify the IPv4 addresses used for translation at the **[edit services nat pool]** hierarchy level. This pool must be referenced in the rule that translates the IPv6 addresses to IPv4.

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

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

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

Configuring NAT Rule Sets

The **rule-set** statement defines a collection of NAT 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. You then specify the order of the rules by

including the **rule-set** statement at the **[edit services nat]** 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 match the packet, no NAT action is performed on the packet. If a packet is destined to a NAT pool address, it is dropped.

Configuring NAT Service Sets

When configuring a service set for NAT processing, make sure you have defined:

- Service interface(s) for handling inbound and outbound traffic

NOTE: Prior to Junos OS Release 11.4R3, you could only use a source NAT pool in a single service set. As of Junos OS Release 11.4R3 and subsequent releases, you can reuse a source NAT pool in multiple service sets, provided that the service interfaces associated with the service sets are in different virtual routing and forwarding (VRF) instances.

- For interface style service sets, when a NAT pool is reused in multiple service sets, the service interfaces used in the **interface-service service-interface** option of each service set must be in different VRFs.
- For next-hop style service sets, when a NAT pool is reused in multiple service sets, the service interfaces used in the **outside-interface** option of each service set must be in different VRFs.

  Not adhering to these service interface restrictions will cause multiple routes to be installed in the same VRF for the same NAT addresses, causing reverse traffic to be processed incorrectly.

To enable sharing of source NAT pools, include the **allow-overlapping-nat-pools** statement at the **[edit services nat]** hierarchy level.

- A NAT rule or ruleset

NOTE: To configure an MX-DPC interface to be used exclusively for carrier-grade NAT (CGN) or related services (intrusion detection, stateful firewall, and softwire), include the cgn-pic statement at the **[edit interfaces** *interface-name* **services-options]** hierarchy level.

To configure a NAT service set:

1. At the **[edit services]** hierarchy level, define the service set.

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

2. Configure either an interface service, which requires a single service interface, or a next-hop service, which requires an inside and outside service interface.

   ```
   [edit services service-set service-set-name]
   user@host# set interface-service service-interface interface-name
   ```

   Or

   ```
   [edit services service-set service-set-name]
   user@host# set next-hop-service inside-service-interface interface-name
     outside-service-interface interface-name
   ```

   NOTE: If you have a Trio-based line card (MPC/MIC), you can use an inline-services interface that was configured on that card, as shown in this example:

   ```
   [edit]
   user@host# set interfaces si-0/0/0
   [edit services service-set s1]
   user@host# set interface-service service-interface si-0/0/0
   ```

   For more information on interface service and next-hop service, see “Configuring Service Sets to be Applied to Services Interfaces”.

3. Configure a reference to the NAT rules or ruleset to be used with the service set.

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

4. (Optional) For NAT64, specify that the don’t fragment (DF) bit for IPv4 packet headers is cleared when packet length is less than 1280 bytes.

   ```
   [edit services service-set service-set-name]
   user@host# set nat-options stateful-nat64 clear-dont-fragment-bit
   ```

Related Documentation

- Configuring Service Sets to be Applied to Services Interfaces

Configuring Static Source Translation in IPv4 Networks

To configure the translation type as **basic-nat44**, you must configure the NAT pool and rule, service set with service interface, and trace options. This topic includes the following tasks:

1. Configuring the NAT Pool and Rule on page 38
2. Configuring the Service Set for NAT on page 39
3. Configuring Trace Options on page 40

Configuring the NAT Pool and Rule

To configure the NAT pool, rule, and term:

1. In configuration mode, go to the **[edit services nat]** hierarchy level.

   ```
   [edit]
   user@host# edit services nat
   ```

2. Configure the NAT pool with an address.

   ```
   [edit services nat]
   user@host# set pool pool name address address
   ```

   In the following example, the pool name is **src_pool** and the address is **10.10.10.2/32**.

   ```
   [edit services nat]
   user@host# set pool src_pool address 10.10.10.2/32
   ```

3. Configure the NAT rule and the match direction.

   ```
   [edit services nat]
   user@host# set rule rule-name match-direction match-direction
   ```

   In the following example, the NAT rule name is **rule-basic-nat44** and the match direction is **input**.

   ```
   [edit services nat]
   user@host# set rule rule-basic-nat44 match-direction input
   ```

4. Configure the source address in the **from** statement.

   ```
   [edit services nat]
   user@host# set rule rule-basic-nat44 term term-name from from
   ```

   In the following example, the term name is **t1** and the input condition is **source-address 3.1.1.2/32**.

   ```
   [edit services nat]
   user@host# set rule rule-basic-nat44 term t1 from source-address 3.1.1.2/32
   ```

5. Configure the NAT term action and properties of the translated traffic.

   ```
   [edit services nat]
   user@host# set rule rule-basic-nat44 term t1 then term-action translated-property
   ```

   In the following example, the term action is **translated** and the property of the translated traffic is **source-pool src_pool**.

   ```
   [edit services nat]
   user@host# set rule rule-basic-nat44 term t1 then translated source-pool src_pool
   ```

6. Configure the translation type.

   ```
   [edit services nat]
   user@host# set rule rule-basic-nat44 term t1 then translated translation-type
     translation-type
   ```

   In the following example, the translation type is **basic-nat44**.

   ```
   [edit services nat]
   ```

```
user@host# set rule rule-basic-nat44 term t1 then translated translation-type
  basic-nat44
```

7. Verify the configuration by using the **show** command at the **[edit services nat]** hierarchy level.

```
[edit services]

user@host# show

nat {
    pool src_pool {
        address 10.10.10.2/32;
    }
    rule rule-basic-nat44 {
        match-direction input;
        term t1 {
            from {
                source-address {
                    3.1.1.2/32;
                }
            }
            then {
                translated {
                    source-pool src_pool;
                    translation-type {
                        basic-nat44;
                    }
                }
            }
        }
    }
}
```

Configuring the Service Set for NAT

To configure the service set for NAT:

1. In configuration mode, go to the **[edit services]** hierarchy level.

```
[edit]
user@host# edit services
```

2. Configure the service set.

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

In the following example, the service set name is **s1**.

```
[edit services]
user@host# edit service-set s1
```

3. For the **s1** service set, set the reference to the NAT rules configured at the **[edit services nat]** hierarchy level.

```
[edit services service-set s1]
user@host# set nat-rules rule-name
```

In the following example, the rule name is **rule-basic-nat44**.

```
[edit services service-set s1]
```

```
user@host# set nat-rules rule-basic-nat44
```

4. Configure the service interface.

```
[edit services service-set s1]
user@host# set interface-service service-interface service-interface-name
```

In the following example, the service interface name is **ms-1/2/0**.

```
[edit services service-set s1]
user@host# set interface-service service-interface ms-1/2/0
```

NOTE: If you have a Trio-based line card, you can configure an inline-services interface on that card:

```
[edit]
user@host# set interfaces si-0/0/0
[edit services service-set s1]
user@host# set interface-service service-interface si-0/0/0
```

5. Verify the configuration by using the **show** command at the **[edit services]** hierarchy level.

```
[edit services]
user@host# show
service-set s1 {
    nat-rules rule-basic-nat44;
    interface-service {
        service-interface ms-1/2/0;
    }
}
```

Configuring Trace Options

To configure the trace options:

1. In configuration mode, go to the **[edit services adaptive-services-pics]** hierarchy level.

```
[edit]
user@host# edit services adaptive-services-pics
```

2. Configure the trace options.

```
[edit services adaptive-services-pics]
user@host# set traceoptions flag tracing parameter
```

In the following example, the tracing parameter is **all**.

```
[edit services adaptive-services-pics]
user@host# set traceoptions flag all
```

3. Verify the configuration by using the **show** command at the **[edit services]** hierarchy level.

```
[edit services]
user@host# show
adaptive-services-pics {
    traceoptions {
```

```
        flag all;
    }                                   }
```

Configuring Static Source Translation in IPv6 Networks

To configure the translation type as **basic-nat66**, you must configure the NAT pool and rule, service set with service interface, and trace options. This topic includes the following tasks:

1. Configuring the NAT Pool and Rule on page 41
2. Configuring the Service Set for NAT on page 42
3. Configuring Trace Options on page 43

Configuring the NAT Pool and Rule

To configure the NAT pool, rule, and term:

1. In configuration mode, go to the **[edit services nat]** hierarchy level.

   ```
   [edit]
   user@host# edit services nat
   ```

2. Configure the NAT pool with an address.

   ```
   [edit services nat]
   user@host# set pool pool name address address
   ```

   In the following example, the pool name is **src_pool** and the address is **10.10.10.2/32**.

   ```
   [edit services nat]
   user@host# set pool src_pool address 10.10.10.2/32
   ```

3. Configure the NAT rule and the match direction.

   ```
   [edit services nat]
   user@host# set rule rule-name match-direction match-direction
   ```

   In the following example, the rule name is **rule-basic-nat66** and the match direction is **input**.

   ```
   [edit services nat]
   user@host# set rule rule-basic-nat66 match-direction input
   ```

4. Configure the source address in the **from** statement.

   ```
   [edit services nat]
   user@host# set rule rule-basic-nat66 term term-name from from
   ```

   In the following, the term name is **t1** and the input condition is **source-address 10:10:10::0/96**.

   ```
   [edit services nat]
   user@host# set rule rule-basic-nat66 term t1 from source-address 10:10:10::0/96
   ```

5. Configure the NAT term action and properties of the translated traffic.

   ```
   [edit services nat]
   user@host# set rule rule-basic-nat66 term t1 then term-action translated-property
   ```

In the following example, the term action is **translated** and the property of the translated traffic is **source-pool src_pool**.

```
[edit services nat]
user@host# set rule rule-basic-nat66 term t1 then translated source-pool src_pool
```

6. Configure the translation type.

```
[edit services nat]
user@host# set rule rule-basic-nat66 term t1 then translated translation-type
  translation-type
```

In the following example, the translation type is **basic-nat66**.

```
[edit services nat]
user@host# set rule rule-basic-nat66 term t1 then translated translation-type
  basic-nat66
```

7. Verify the configuration by using the **show** command at the **[edit services]** hierarchy level.

```
[edit services]
user@host# show
nat {
    pool src_pool {
        address 10.10.10.2/32;
    }
    rule rule-basic-nat66 {
        match-direction input;
        term t1 {
            from {
                source-address {
                    10:10:10::0/96;
                }
            }
            then {
                translated {
                    source-pool src_pool;
                    translation-type {
                        basic-nat66;
                    }
                }
            }
        }
    }
}
```

Configuring the Service Set for NAT

To configure the service set for NAT:

1. In configuration mode, go to the **[edit services]** hierarchy level.

```
[edit]
user@host# edit services
```

2. Configure the service set.

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

In the following example, the service set name is **s1**.

```
[edit services]
user@host# edit service-set s1
```

3. For the **s1** service set, set the reference to the NAT rules configured at the **[edit services nat]** hierarchy level.

```
[edit services service-set s1]
user@host# set nat-rules rule-name
```

In the following example, the rule name is **rule-basic-nat66**.

```
[edit services service-set s1]
user@host# set nat-rules rule-basic-nat66
```

4. Configure the service interface.

```
[edit services service-set s1]
user@host# set interface-service service-interface service-interface-name
```

In the following example, the service interface name is **sp-1/2/0**.

```
[edit services service-set s1]
user@host# set interface-service service-interface sp-1/2/0
```

5. Verify the configuration by using the **show** command at the **[edit services]** hierarchy level.

```
[edit services]
user@host# show
service-set s1 {
    nat-rules rule-basic-nat66;
    interface-service {
        service-interface sp-1/2/0;
    }
}
```

Configuring Trace Options

To configure the trace options at the **[edit services adaptive-services-pics]** hierarchy level:

1. In configuration mode, go to the **[edit services adaptive-services-pics]** hierarchy level.

```
[edit]
user@host# edit services adaptive-services-pics
```

2. Configure the trace options.

```
[edit services adaptive-services-pics]
user@host# set traceoptions flag tracing parameter
```

In the following example, the tracing parameter is **all**.

```
[edit services adaptive-services-pics]
user@host# set traceoptions flag all
```

3. Verify the configuration by using the **show** command at the **[edit services]** hierarchy level.

```
[edit services]
user@host# show
```

```
adaptive-services-pics {
    traceoptions {
        flag all;
    }
}
```

Configuring Dynamic Source Address and Port Translation in IPv4 Networks

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

To configure NAPT, you must configure a rule at the **[edit services nat]** hierarchy level for dynamically translating the source IPv4 addresses.

To configure the NAPT in IPv4 networks:

1. In configuration mode, go to the **[edit services]** hierarchy level.

   ```
   [edit]
   user@host# edit services
   ```

2. Configure the service set and NAT rule.

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

   In the following example, the name of the service set is **s1** and the name of the NAT rule is **rule-napt-44**.

   ```
   [edit services]
   user@host# set service-set s1 nat-rules rule-napt-44
   ```

3. Go to the **[interface-service]** hierarchy level of the service set.

   ```
   [edit services]
   user@host# edit service-set s1 interface-service
   ```

4. Configure the service interface.

   ```
   [edit services service-set s1 interface service]
   user@host# set service-interface service-interface-name
   ```

   In the following example, the name of the service interface is **ms-0/1/0**.

   NOTE: If the service interface is not present in the router, or the specified interface is not functional, the following command can result in an error.

   ```
   [edit services service-set s1 interface service]
   user@host# set service-interface ms-0/1/0
   ```

5. Go to the **[edit services nat]** hierarchy level. Issue the command from the top of the services hierarchy, or use the **top** keyword.

   ```
   [edit services service-set s1 interface service]
   user@host# top edit services nat
   ```

6. Configure the NAT pool with an address.

   ```
   [edit services nat]
   user@host# set pool pool-name address address
   ```

   In the following example, the name of the pool is **napt-pool** and the address is **10.10.10.0**.

   ```
   [edit services nat]
   user@host# set pool napt-pool address 10.10.10.0
   ```

7. Configure the port.

   ```
   [edit services nat]
   user@host# set pool pool-name port port-type
   ```

   In the following example, the port type is selected as **automatic**.

   ```
   [edit services nat]
   user@host# set pool napt-pool port automatic
   ```

8. Configure the rule and the match direction.

   ```
   [edit services nat]
   user@host# set rule rule-name match-direction match-direction
   ```

   In the following example, the name of the rule is **rule-napt-44** and the match direction is **input**.

   ```
   [edit services nat]
   user@host# set rule rule-napt-44 match-direction input
   ```

9. Configure the term, the action for the translated traffic, and the translation type.

   ```
   [edit services nat]
   user@host# set rule rule-name term term-name then translated translated-action
     translation-type translation- type
   ```

   In the following example, the name of the term is **t1**, the action for the translated traffic is **translated**, the name of the source pool is **napt-pool**, and the translation type is **napt-44**.

   ```
   [edit services nat]
   user@host# set rule rule-napt-44 match-direction input term t1 then translated
     source-pool napt-pool translation-type napt-44
   ```

10. Go to the **[edit services adaptive-services-pics]** hierarchy level. In the command, the **top** keyword ensures that the command is run from the top of the hierarchy.

    ```
    [edit services nat]
    user@host# top edit services adaptive-services-pics
    ```

11. Configure the trace options.

    ```
    [edit services adaptive-services-pics]
    user@host# set traceoptions flag tracing parameter
    ```

    In the following example, the tracing parameter is configured as **all**.

    ```
    [edit services adaptive-services-pics]
    ```

```
user@host# set traceoptions flag all
```

12. Verify the configuration by using the **show** command at the **[edit services]** hierarchy level.

```
[edit services]
user@host# show
service-set s1 {
    nat-rules rule-napt-44;
    interface-service {
        service-interface ms-0/1/0;
    }
}
nat {
    pool napt-pool {
        address 10.10.10.0/32;
        port {
            automatic;
        }
    }
    rule rule-napt-44 {
        match-direction input;
        term t1 {
            then {
                translated {
                    source-pool napt-pool;
                    translation-type {
                        napt-44;
                    }
                }
            }
        }
    }
}
adaptive-services-pics {
    traceoptions {
        flag all;
    }
}
```

Related Documentation

- Example: Configuring Dynamic Source Translation for an IPv4 Network on page 78

Configuring Dynamic Source Address and Port Translation for IPv6 Networks

Network Address Port Translation (NAPT) is a method by which many network addresses and their TCP/UDP ports are translated into a single network address and its TCP/UDP ports. This translation can be configured in both IPv4 and IPv6 networks. This section describes the steps for configuring NAPT in IPv6 networks. For information about configuring NAPT in IPv4 networks, see "Configuring Dynamic Source Address and Port Translation in IPv4 Networks" on page 44.

To configure NAPT, you must configure a rule at the **[edit services nat]** hierarchy level for dynamically translating the source IPv6 addresses.

To configure NAPT in IPv6 networks:

1. In configuration mode, go to the **[edit services nat]** hierarchy level.

   ```
   [edit]
   user@host# edit services nat
   ```

2. Define the pool of IPv6 source addresses that must be used for dynamic translation. For NAPT, also specify port numbers when configuring the source pool.

   ```
   [edit services nat]
   user@host# set pool pool name address IPv6 source addresses
   user@host# set pool pool name port source ports
   ```

   For example:

   ```
   [edit services nat]
   user@host#  set pool IPV6-NAPT-Pool address 2002::1/96
   user@host#  set pool IPV6-NAPT-Pool port automatic
   ```

3. Define a NAT rule for translating the source addresses. To do this, set the **match-direction** statement of the rule as **input**. In addition, define a term that uses **napt-66** as the translation type for translating the addresses of the pool defined in the previous step.

   ```
   [edit services nat]
   user@host# set rule rule name match-direction input
   user@host# set rule rule name term term name then translated source-pool pool name
   user@host# set rule rule name term term name then translated translation-type napt-66
   ```

   For example:

   ```
   [edit services nat]
   user@host# set rule IPV6-NAPT-Rule match-direction input
   user@host# set rule IPV6-NAPT-Rule term t1 then translated source-pool
     IPV6-NAPT-Pool
   user@host# set rule IPV6-NAPT-Rule term t1 then translated translation-type napt-66
   ```

4. Enter the **up** command to navigate to the **[edit services]** hierarchy level.

   ```
   [edit services nat]
   user@host# up
   ```

5. Define a service set to specify the services interface that must be used, and reference the NAT rule implemented for NAPT translation.

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

For example:

```
[edit services]
user@host# set service-set IPV6-NAPT-ServiceSet interface- service service-interface
  ms-0/1/0
user@host# set service-set IPV6-NAPT-ServiceSet nat-rules IPV6-NAPT-Rule
```

6. Define the trace options for the adaptive services PIC.

```
[edit services]
user@host# set adaptive-services-pics traceoptions flag tracing parameter
```

For example:

```
[edit services]
user@host# set adaptive-services-pics traceoptions flag all
```

Related Documentation

- Example: Configuring Dynamic Source Address and Port Translation for an IPv6 Network

Configuring Dynamic Address-Only Source Translation in IPv4 Networks

In IPv4 networks, dynamic address translation (dynamic NAT) is a mechanism to dynamically translate the destination traffic without port mapping. To use dynamic NAT, you must specify a source pool name, which includes an address configuration.

To configure dynamic NAT in IPv4 networks:

1. In configuration mode, go to the **[edit services]** hierarchy level.

```
[edit]
user@host# edit services
```

2. Configure the service set and NAT rule.

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

In the following example, the name of the service set is **s1**, and the name of the NAT rule is **rule-dynamic-nat44**.

```
[edit services]
user@host# set service-set s1 nat-rules rule-dynamic-nat44
```

3. Go to the **[interface-service]** hierarchy level for the service set.

```
[edit services]
user@host# edit service-set s1 interface-service
```

4. Configure the service interface.

```
[edit services service-set s1 interface-service]
user@host# set service-interface service-interface-name
```

In the following example, the name of the service interface is **ms-0/1/0**.

NOTE: If the service interface is not present in the router, or the specified interface is not functional, the following command can result in an error.

```
[edit services service-set s1 interface-service]
user@host# set service-interface ms-0/1/0
```

5. Go to the **[edit services nat]** hierarchy level. Issue the following command from the top of the services hierarchy, or use the **top** keyword.

   ```
   [edit services service-set s1 interface-service]
   user@host# top edit services nat
   ```

6. Configure the NAT pool with an address.

   ```
   [edit services nat]
   user@host# set pool pool-name address address
   ```

   In the following example, the name of the pool is **source-dynamic-pool**, and the address is **10.10.10.0**.

   ```
   [edit services nat]
   user@host# set pool source-dynamic-pool address 10.10.10.0
   ```

7. Configure the rule, match direction, term, and source address.

   ```
   [edit services nat]
   user@host# set rule rule-name match-direction match-direction term term-name from
     source-address address
   ```

   In the following example, the name of the rule is **rule-dynamic-nat44**, the match direction is **input**, the name of the term is **t1**, and the source address is **3.1.1.0**.

   ```
   [edit services nat]
   user@host# set rule rule-dynamic-nat44 match-direction input term t1 from
     source-address 3.1.1.0
   ```

8. Go to the **[edit rule rule-dynamic-nat-44 term t1]** hierarchy level.

   ```
   [edit services nat]
   user@host# edit rule rule-dynamic-nat44 term t1
   ```

9. Configure the source pool and the translation type.

   ```
   [edit services nat rule rule-dynamic-nat44 term t1]
   user@host# set then translated source-pool src-pool-name translation-type
     translation-type
   ```

   In the following example, the name of the source pool is **source-dynamic-pool** and the translation type is **dynamic-nat44**.

   ```
   [edit services nat rule rule-dynamic-nat44 term t1]
   user@host# set then translated source-pool  source-dynamic-pool translation-type
     dynamic-nat44
   ```

10. Go to the **[edit services adaptive-services-pics]** hierarchy level. In the following command, the **top** keyword ensures that the command is run from the top of the hierarchy.

    ```
    [edit services nat rule rule-dynamic-nat44 term t1]
    ```

```
user@host# top edit services adaptive-services-pics
```

11. Configure the trace options.

```
[edit services adaptive-services-pics]
user@host# set traceoptions flag tracing parameter
```

In the following example, the tracing parameter is configured as **all**.

```
[edit services adaptive-services-pics]
user@host# set traceoptions flag all
```

12. Verify the configuration by using the **show** command at the **[edit services]** hierarchy level.

```
[edit services]
user@host# show
service-set s1 {
    nat-rules rule-dynamic-nat44;
    interface-service {
        service-interface ms-0/1/0;
    }
}
 nat {
    pool source-dynamic-pool {
        address 10.1.1.0/24;
    }
    rule rule-dynamic-nat44 {
        match-direction input;
        term t1 {
            from {
                source-address {
                    3.1.1.0/24;
                }
            }
            then {
                translated {
                    destination-pool source-dynamic-pool;
                    translation-type {
                        dynamic-nat44;
                    }
                }
            }
        }
    }
}
 adaptive-services-pics {
    traceoptions {
        flag all;
    }
  }
```

Related Documentation

- Example: Configuring Dynamic Address-Only Source Translation in an IPv4 Network on page 79

Configuring Static Destination Address Translation in IPv4 Networks

In IPv4 networks, destination address translation is a mechanism used to implement address translation for destination traffic without port mapping. To use destination address translation, the size of the pool address space must be greater than or equal to the destination address space. You must specify a name for the **destination-pool** statement, which can contain multiple addresses, ranges, or prefixes, as long as the number of NAT addresses in the pool is larger than the number of destination addresses in the **from** statement.

To configure destination address translation in IPv4 networks:

1. In configuration mode, go to the **[edit services]** hierarchy level.

   ```
   [edit]
   user@host# edit services
   ```

2. Configure the service set and the NAT rule.

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

   In the following example, the name of the service set is **s1** and the name of the NAT rule is **rule-dnat44**.

   ```
   [edit services]
   user@host# set service-set s1 nat-rules rule-dnat44
   ```

3. Go to the **[interface-service]** hierarchy level of the service set.

   ```
   [edit services]
   user@host# edit service-set s1 interface-service
   ```

4. Configure the service interface.

   ```
   [edit services service-set s1 interface-service]
   user@host# set service-interface service-interface-name
   ```

   In the following example, the name of the service interface is **ms-0/1/0**.

   NOTE: If the service interface is not present in the router, or the specified interface is not functional, the following command can result in an error.

   ```
   [edit services service-set s1 interface-service]
   user@host# set service-interface ms-0/1/0
   ```

5. Go to the **[edit services nat]** hierarchy level. Issue the following command from the top of the services hierarchy, or use the **top** keyword.

   ```
   [edit services service-set s1]
   user@host# top edit services nat
   ```

6. Configure the NAT pool with an address.

   ```
   [edit services nat]
   user@host# set pool pool-name address address
   ```

In the following example, **dest-pool** is used as the pool name and **4.1.1.2** as the address.

```
user@host# set pool dest-pool address 4.1.1.2
```

7. Configure the rule, match direction, term, and destination address.

```
[edit services nat]
user@host# set rule rule-name match-direction match-direction term term-name from
  destination-address address
```

In the following example, the name of the rule is **rule-dnat44**, the match direction is **input**, the name of the term is **t1**, and the address is **20.20.20.20**.

```
[edit services nat]
user@host# set rule rule-dnat44 match-direction input term t1 from
  destination-address 20.20.20.20
```

8. Go to the **[edit services nat rule rule-dnat44 term t1]** hierarchy level.

```
[edit services nat]
user@host# edit rule rule-dnat44 term t1
```

9. Configure the destination pool and the translation type.

```
[edit services nat rule rule-dnat44 term t1]
user@host# set then translated destination-pool dest-pool-name translation-type
  translation-type
```

In the following example, the destination pool name is **dest-pool**, and the translation type is **dnat-44**.

```
[edit services nat rule rule-dnat44 term t1]
user@host# set then translated destination-pool dest-pool translation-type dnat-44
```

10. Go to the **[edit services adaptive-services-pics]** hierarchy level. In the following command, the **top** keyword ensures that the command is run from the top of the hierarchy.

```
[edit services nat rule rule-dnat44 term t1]
user@host# top edit services adaptive-services-pics
```

11. Configure the trace options.

```
[edit services adaptive-services-pics]
user@host# set traceoptions flag tracing parameter
```

In the following example, the tracing parameter is configured as **all**.

```
[edit services adaptive-services-pics]
user@host# set traceoptions flag all
```

12. Verify the configuration by using the **show** command at the **[edit services]** hierarchy level.

```
[edit services]
user@host# show
service-set s1 {
    nat-rules rule-dnat44;
    interface-service {
        service-interface ms-0/1/0;
    }
}
 nat {
```

```
        pool dest-pool {
            address 4.1.1.2/32;
        }
        rule rule-dnat44 {
            match-direction input;
            term t1 {
                from {
                    destination-address {
                        20.20.20.20/32;
                    }
                }
                then {
                    translated {
                        destination-pool dest-pool;
                        translation-type {
                            dnat-44;
                        }
                    }
                }
            }
        }
    }
    adaptive-services-pics {
        traceoptions {
            flag all;
        }
    }
```

Related Documentation

- Example: Configuring Static Destination Address Translation on page 80

Configuring Translation Type for Translation Between IPv6 and IPv4 Networks

To configure the translation type as **basic-nat-pt**, you must configure the DNS ALG application, the NAT pools and rules, a service set with a service interface, and trace options. This topic includes the following tasks:

1. Configuring the DNS ALG Application on page 53
2. Configuring the NAT Pool and NAT Rule on page 54
3. Configuring the Service Set for NAT on page 57
4. Configuring Trace Options on page 58

Configuring the DNS ALG Application

To configure the DNS ALG application:

1. In configuration mode, go to the **[edit applications]** hierarchy level.

   ```
   [edit]
   user@host# edit applications
   ```

2. Configure the ALG to which the DNS traffic is destined at the **[edit applications]** hierarchy level. Define the application name and specify the application protocol to use in match conditions in the first NAT rule or term.

   ```
   [edit applications]
   ```

user@host# set application *application-name* application-protocol *application-protocol*

In the following example, the application name is **dns-alg** and application protocol is **dns**.

```
[edit applications]
user@host# set application dns-alg application-protocol dns
```

3. Verify the configuration by using the **show** command at the **[edit applications]** hierarchy level.

```
[edit applications]
user@host# show
application dns-alg {
    application-protocol dns;
}
```

Configuring the NAT Pool and NAT Rule

To configure the NAT pool and NAT rule:

1. In configuration mode, go to the **[edit services nat]** hierarchy level.

```
[edit]
user@host# edit services nat
```

2. Configure the NAT pool and its address.

```
[edit services nat]
user@host# set pool pool-name address address
```

In the following example, the name of the NAT pool is **p1** and the address is **10.10.10.2/32**.

```
[edit services nat]
user@host# set pool p1 address 10.10.10.2/32
```

3. Configure the source pool and its address.

```
[edit services nat]
user@host# set pool source-pool-name address address
```

In the following example, the name of the source pool is **src_pool0** and the source pool address is **20.1.1.1/32**.

```
[edit services nat]
user@host# set pool src_pool0 address 20.1.1.1/32
```

4. Configure the destination pool and its address.

```
[edit services nat]
user@host# set pool destination-pool-name address address
```

In the following example, the name of the destination pool is **dst_pool0** and the destination pool address is **50.1.1.2/32**.

```
[edit services nat]
user@host# set pool dst_pool0 address 50.1.1.2/32
```

5. Configure the rule and the match direction.

```
[edit services nat]
```

```
user@host# set rule rule-name match-direction match-direction
```

In the following example, the rule name is **rule-basic-nat-pt** and the match direction is **input**.

```
[edit services nat]
user@host# set rule basic-nat-pt match-direction input
```

6. Configure the term and the input conditions for the NAT term.

```
[edit services nat]
user@host# set rule rule-basic-nat-pt term term from from
```

In the following example, the term is **t1** and the input conditions are **source-address 2000::2/128**, **destination-address 4000::2/128**, and **applications dns_alg**.

```
[edit services nat]
user@host# set rule rule-basic-nat-pt term t1 from source-address 2000::2/128
[edit services nat]
user@host# set rule rule-basic-nat-pt term t1 from destination-address 4000::2/128
[edit services nat]
user@host# set rule rule-basic-nat-pt term t1 from applications dns_alg
```

7. Configure the NAT term action and the properties of the translated traffic.

```
[edit services nat]
user@host# set rule rule-basic-nat-pt term t1 then term-action translated-property
```

In the following example, the term action is **translated** and the properties of the translated traffic are **source-pool src_pool0**, **destination-pool dst_pool0**, and **dns-alg-prefix 10:10:10::0/96**.

```
[edit services nat]
user@host# set rule rule-basic-nat-pt term t1 then translated source-pool src_pool0
[edit services nat]
user@host# set rule rule-basic-nat-pt term t1 then translated destination-pool
  dst_pool0
[edit services nat]
user@host# set rule rule-basic-nat-pt term t1 then translated dns-alg-prefix
  10:10:10::0/96
```

8. Configure the translation type.

```
[edit services nat]
user@host# set rule rule-basic-nat-pt term t1 then translated translation-type
  translation-type
```

In the following example, the translation type is **basic-nat-pt**.

```
[edit services nat]
user@host# set rule rule-basic-nat-pt term t1 then translated translation-type
  basic-nat-pt
```

9. Configure another term and the input conditions for the NAT term.

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

In the following example, the term name is **t2** and the input conditions are **source-address 2000::2/128** and **destination-address 10:10:10::0/96**.

```
[edit services nat]
```

```
user@host# set rule rule-basic-nat-pt term t2 from source-address 2000::2/128
[edit services nat]
user@host# set rule rule-basic-nat-pt term t2 from destination-address 10:10:10::0/96
```

10. Configure the NAT term action and the property of the translated traffic.

```
[edit services nat]
user@host# set rule rule-basic-nat-pt term t2 then term-action translated-property
```

In the following example, the term action is **translated** and the property of the translated traffic is **source-prefix 19.19.19.1/32**.

```
[edit services nat]
user@host# set rule rule-basic-nat-pt term t2 then translated source-prefix
  19.19.19.1/32
```

11. Configure the translation type.

```
[edit services nat]
user@host# set rule rule-basic-nat-pt term t2 then translated translation-type
  translation-type
```

In the following example, the translation type is **basic-nat-pt**.

```
[edit services nat]
user@host# set rule rule-basic-nat-pt term t2 then translated translation-type
  basic-nat-pt
```

12. Verify the configuration by using the **show** command at the **[edit services nat]** hierarchy level.

```
[edit services nat]
user@host# show
pool p1 {
    address 10.10.10.2/32;
}
pool src_pool0 {
    address 20.1.1.1/32;
}
pool dst_pool0 {
    address 50.1.1.2/32;
}
rule rule-basic-nat-pt {
    match-direction input;
    term t1 {
        from {
            source-address {
                2000::2/128;
            }
            destination-address {
                4000::2/128;
            }
            applications dns_alg;
        }
        then {
            translated {
                source-pool src_pool0;
                destination-pool dst_pool0;
                dns-alg-prefix 10:10:10::0/96;
                translation-type {
                    basic-nat-pt;
                }
```



```
                }
            }
        }
        term t2 {
            from {
                source-address {
                    2000::2/128;
                }
                destination-address {
                    10:10:10::0/96;
                }
            }
            then {
                translated {
                    source-prefix 19.19.19.1/32;
                    translation-type {
                        basic-nat-pt;
                    }
                }
            }
        }
    }
```

Configuring the Service Set for NAT

To configure the service set for NAT:

1. In configuration mode, go to the **[edit services]** hierarchy level.

   ```
   [edit]
   user@host# edit services
   ```

2. Configure the service set.

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

   In the following example, the name of the service set is **ss_dns**.

   ```
   [edit services]
   user@host# edit service-set ss_dns
   ```

3. Configure the service set with NAT rules.

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

   In the following example, the rule name is **rule-basic-nat-pt**.

   ```
   [edit services service-set ss_dns]
   user@host# set nat-rules rule-basic-nat-pt
   ```

4. Configure the service interface.

   ```
   [edit services service-set ss_dns]
   user@host# set interface-service service-interface service-interface-name
   ```

   In the following example, the name of service interface is **sp-1/2/0**.

   ```
   [edit services service-set ss_dns]
   ```

user@host# set interface-service service-interface sp-1/2/0

5. Verify the configuration by using the **show services** command from the **[edit]** hierarchy level.

```
[edit]
user@host# show services
    service-set ss_dns {
        nat-rules rule-basic-nat-pt;
        interface-service {
            service-interface sp-1/2/0;
        }
    }
```

Configuring Trace Options

To configure the trace optionsl:

1. In configuration mode, go to the **[edit services adaptive-services-pics]** hierarchy level.

   [edit]
   user@host# edit services adaptive-services-pics

2. Configure the trace options.

   [edit services adaptive-services-pics]
   user@host# set traceoptions flag *tracing parameter*

   In the following example, the tracing parameter is **all**.

   [edit services adaptive-services-pics]
   user@host# set traceoptions flag all

3. Verify the configuration by using the **show** command at the **[edit services]** hierarchy level.

```
[edit services]
user@host# show
adaptive-services-pics {
    traceoptions {
        flag all;
    }
}
```

Configuring NAT-PT

To configure Network Address Translation–Protocol Translation (NAT-PT), you must configure a Domain Name System application-level gateway (DNS ALG) application to map addresses returned in the DNS response to an IPv6 address. DNS ALG is used with NAT-PT to facilitate name-to-address mapping. When configuring NAT-PT, network address translation can either be an address-only translation or an address and port translation. The Junos OS implementation is described in RFC 2766 and RFC 2694.

Before you begin configuring NAT-PT with DNS ALG, you must have the following configured:

- NAT with two rules or one rule and two terms. The first NAT rule or term ensures that the DNS query and response packets are translated correctly. For this rule to work, you must configure a DNS ALG application and reference it in the first rule. The second rule or term is required to ensure that NAT sessions are destined to the address mapped by the DNS ALG application.
- A service set that references the first NAT rule or term and a multiservices interface.

To configure NAT-PT with DNS ALG:

1. Configure the DNS session that processes packets to the DNS server:

   a. Configure the ALG to which the DNS traffic is destined at the **[edit applications]** hierarchy level. Define the application name and specify the application protocol to use in match conditions in the first NAT rule or term.

      [edit applications]
      user@host# **set application** *application-name* **application-protocol** *application-protocol*

      For example:

      [edit applications]
      user@host# **set application dns_alg application-protocol dns**

   b. Reference the ALG in the first NAT rule or term.

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

      In the following example, the application name is **dns_alg**.

      [edit services nat rule rule1 term term1]
      user@host# **set from applications dns_alg**

   c. Define the DNS ALG pool or prefix for mapping IPv4 addresses to IPv6 addresses.

      [edit services nat rule *rule-name* term *term-name*]
      user@host# **set then translated dns-alg-prefix** *dns-alg-prefix*
      user@host# **set then translated dns-alg-pool** *dns-alg-pool*

      The following example shows the configuration of the 96-bit prefix for mapping IPv4 address to IPv6 addresses.

      [edit services nat rule rule1 term term1]
      user@host# **set then translated dns-alg-prefix 10:10:10::0/96**

The following sample output shows the minimum configuration of the application.

```
[edit applications]
user@host# show
application dns_alg {
    application-protocol dns;
}
```

The following sample output shows the minimum configuration of the first NAT rule.

```
[edit services nat]
user@host# show
```

```
rule rule1 {
            applications dns_alg;
        }
        then {
            translated {
                dns-alg-prefix 10:10:10::0/96;
                }
            }
        }
    }
}
```

The following sample output shows the minimum configuration of the second NAT rule.

```
[edit services nat]
user@host# show
rule rule2 {
    term term1 {
        from {
            destination-address {
                10:10:10::c0a8:108/128;
            }
        }
        then {
            translated {
                source-prefix 19.19.19.1/32;
                }
            }
        }
    }
}
```

Related Documentation

- Network Address Translation Overview on page 7
- Example: Configuring NAT-PT on page 86
- dns-alg-prefix on page 133
- dns-alg-pool on page 132

Example: Configuring Port Forwarding with Twice NAT

The following example configures port forwarding with **twice-napt-44** as the translation type. The example also has stateful firewall and multiple port maps configured.

```
[edit services]
user@host# show
service-set in {
    syslog {
        host local {
            services any;
        }
    }
    stateful-firewall-rules r;
    nat-rules r;
    interface-service {
        service-interface sp-10/0/0.0;
    }
}
```

```
stateful-firewall {
    rule r {
        match-direction input;
        term t {
            from {
                destination-port {
                    range low 1 high 57000;
                }
            }
            then {
                reject;
            }
        }
    }
}
nat {
    pool x {
        address 12.0.0.2/32;
    }
    rule r {
        match-direction input;
        term t {
            from {
                destination-address {
                    14.0.0.2/32;
                }
                destination-port {
                    range low 10 high 20000;
                }
            }
            then {
                port-forwarding-mappings y;
             translated {
                    destination-pool x;
                    translation-type {
                        twice-napt-44;
                    }
                }
            }
        }
    }
    port-forwarding y {
        destined-port 45;
        translated-port 23;
        destined-port 55;
        translated-port 33;
        destined-port 65;
        translated-port 43;
   }
}
adaptive-services-pics {
    traceoptions {
        file sp-trace;
        flag all;
    }
}
```

NOTE:

- Stateful firewall has precedence over port forwarding. In this example, for instance, no traffic destined to any port between 1 and 57000 will be translated.
- Up to 32 port maps can be configured.

Related Documentation

- Configuring Port Forwarding for Static Destination Address Translation on page 66

Configuring NAT-PT

To configure Network Address Translation–Protocol Translation (NAT-PT), you must configure a Domain Name System application-level gateway (DNS ALG) application to map addresses returned in the DNS response to an IPv6 address. DNS ALG is used with NAT-PT to facilitate name-to-address mapping. When configuring NAT-PT, network address translation can either be an address-only translation or an address and port translation. The Junos OS implementation is described in RFC 2766 and RFC 2694.

Before you begin configuring NAT-PT with DNS ALG, you must have the following configured:

- NAT with two rules or one rule and two terms. The first NAT rule or term ensures that the DNS query and response packets are translated correctly. For this rule to work, you must configure a DNS ALG application and reference it in the first rule. The second rule or term is required to ensure that NAT sessions are destined to the address mapped by the DNS ALG application.
- A service set that references the first NAT rule or term and a multiservices interface.

To configure NAT-PT with DNS ALG:

1. Configure the DNS session that processes packets to the DNS server:

   a. Configure the ALG to which the DNS traffic is destined at the **[edit applications]** hierarchy level. Define the application name and specify the application protocol to use in match conditions in the first NAT rule or term.

      [edit applications]
      user@host# **set application** *application-name* **application-protocol** *application-protocol*

      For example:

      [edit applications]
      user@host# **set application dns_alg application-protocol dns**

   b. Reference the ALG in the first NAT rule or term.

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

      In the following example, the application name is **dns_alg**.

      [edit services nat rule rule1 term term1]
      user@host# **set from applications dns_alg**

   c. Define the DNS ALG pool or prefix for mapping IPv4 addresses to IPv6 addresses.

      [edit services nat rule *rule-name* term *term-name*]
      user@host# **set then translated dns-alg-prefix** *dns-alg-prefix*
      user@host# **set then translated dns-alg-pool** *dns-alg-pool*

      The following example shows the configuration of the 96-bit prefix for mapping IPv4 address to IPv6 addresses.

      [edit services nat rule rule1 term term1]
      user@host# **set then translated dns-alg-prefix 10:10:10::0/96**

The following sample output shows the minimum configuration of the application.

```
[edit applications]
user@host# show
application dns_alg {
    application-protocol dns;
}
```

The following sample output shows the minimum configuration of the first NAT rule.

```
[edit services nat]
user@host# show
rule rule1 {
            applications dns_alg;
        }
        then {
            translated {
                dns-alg-prefix 10:10:10::0/96;
                }
            }
        }
```

```
    }
}
```

The following sample output shows the minimum configuration of the second NAT rule.

```
[edit services nat]
user@host# show
rule rule2 {
    term term1 {
        from {
            destination-address {
                10:10:10::c0a8:108/128;
            }
        }
        then {
            translated {
                source-prefix 19.19.19.1/32;
                }
            }
        }
    }
}
```

Related Documentation

- Network Address Translation Overview on page 7
- Example: Configuring NAT-PT on page 86
- dns-alg-prefix on page 133
- dns-alg-pool on page 132

Configuring Dynamic Source Address and Static Destination Address Translation (IPv6 to IPV4)

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

To configure stateful NAT64, you must configure a rule at the **[edit services nat]** hierarchy level for translating the source address dynamically and the destination address statically.

BEST PRACTICE: When you configure the service set that includes your NAT rule, include the **set stateful-nat64 clear-dont-fragment-bit** at the **[edit services service-set** *service-set-name***]** hierarchy level.This clears the DF (don't fragment) bit in order to prevent unnecessary creation of an IPv6 fragmentation header when translating IPv4 packets that are less than 1280 bytes. RFC 6145, *IP/ICMP Translation Algorithm*, provides a full discussion of the use of the DF flag to control generation of fragmentation headers. For more information on service sets for NAT, see "Configuring NAT Service Sets" on page 36.

To configure stateful NAT64:

1. In configuration mode, go to the **[edit services nat]** hierarchy level:

   ```
   [edit]
   user@host# edit services nat
   ```

2. Define the pool of source addresses to be used for dynamic translation.

   ```
   [edit services nat]
   user@host# set pool pool name address source addresses
   user@host# set pool pool name port source ports
   ```

   For example:

   ```
   [edit services nat]
   user@host#  set pool src-pool-nat64 address 203.0.113.0/24
   user@host#  set pool src-pool-nat64 port automatic
   ```

3. Define a NAT rule for translating the source addresses. Set the **match-direction** statement of the rule as **input**. Then define a term that uses **stateful-nat64** as the translation type for translating the addresses of the pool defined in the previous step.

   ```
   [edit services nat]
   user@host# set rule rule name match-direction input
   user@host# set rule rule name term term name from source-address source address
   user@host# set rule rule name term term name from destination-address destination
     address
   user@host# set rule rule name term term name then translated source-pool pool name
   user@host# set rule rule name term term name then translated destination-prefix
     destination prefix
   ```

```
user@host# set rule rule name term term name then translated translation-type
    stateful-nat64
```

For example:

```
[edit services nat]
user@host# set rule stateful-nat64 match-direction input
user@host# set rule stateful-nat64 term t1 from source-address 2001:DB8::0/96
user@host# set rule stateful-nat64 term t1 from destination-address 64:FF9B::/96
user@host# set rule stateful-nat64 term t1 then translated source-pool src-pool-nat64
user@host# set rule stateful-nat64 term t1 then translated destination-prefix
    64:FF9B::/96
user@host# set rule stateful-nat64 term t1 then translated translation-type
    stateful-nat64
```

Related Documentation

- Example: Configuring Dynamic Source Address and Static Destination Address Translation (IPv6 to IPV4) on page 81

Configuring Port Forwarding for Static Destination Address Translation

Starting with Junos OS Release 11.4, you can map an external IP address and port with an IP address and port in a private network. This allows the destination address and port of a packet to be changed to reach the right host in a Network Address Translation (NAT) gateway. The translation facilitates reaching a host within a masqueraded, typically private, network, based on the port number on which the packet was received from the originating host. Port forwarding allows remote computers, such as public machines on the Internet, to connect to a non-standard port (port other than 80) of a specific computer within a private network. An example of this type of destination is the host of a public HTTP server within a private network. Port forwarding is supported only with **dnat-44** and **twice-napt-44** on IPv4 networks. Port forwarding works only with the FTP application-level gateway (ALG). Port forwarding also supports endpoint-independent mapping (EIM), endpoint-independent filltering (EIF), and address pooling paired (APP). Port forwarding has no support for technologies such as IPv6 rapid deployment (6rd) and dual-stack lite (DS-Lite) that offer IPv6 services over IPv4 infrastructure.

To configure destination address translation with port forwarding in IPv4 networks:

1. In configuration mode, go to the **[edit services nat]** hierarchy level.

   ```
   [edit]
   user@host# edit services nat
   ```

2. Configure the NAT pool with an address.

   ```
   [edit services nat]
   user@host# set pool pool-name address address
   ```

   In the following example, **dest-pool** is used as the pool name and **4.1.1.2** as the address.

   ```
   user@host# set pool dest-pool address 4.1.1.2
   ```

3. Configure the rule, match direction, term, and destination address.

   ```
   [edit services nat]
   ```

```
user@host# set rule rule-name match-direction match-direction term term-name from
  destination-address  address
```

In the following example, the name of the rule is **rule-dnat44**, the match direction is **input**, the name of the term is **t1**, and the address is **20.20.20.20**.

```
[edit services nat]
user@host# set rule rule-dnat44 match-direction input term t1 from
  destination-address 20.20.20.20
```

4. Configure the destination port range.

```
[edit services nat]
user@host# set rule rule-name match-direction match-direction term term-name from
  destination-port range range high | low
```

In the following example, the upper port range is **50** and the lower port range is **20**.

```
[edit services nat]
user@host# set rule rule-dnat44 match-direction input term t1 from destination-port
  range range high 50 low 20
```

5. Go to the **[edit services nat rule rule-dnat44 term t1]** hierarchy level.

```
[edit services nat]
user@host# edit rule rule-dnat44 term t1
```

6. Configure the destination pool.

```
[edit services nat rule rule-dnat44 term t1]
user@host# set then translated destination-pool dest-pool-name
```

In the following example, the destination pool name is **dest-pool**, and the translation type is **dnat-44**.

```
[edit services nat rule rule-dnat44 term t1]
user@host# set then translated destination-pool dest-pool
```

7. Configure the mapping for port forwarding and the translation type.

```
[edit services nat rule rule-dnat44 term t1]
user@host# set then port-forwarding-mappings  map-name translation-type
  translation-type
```

In the following example, the port forwarding map name is **map1**, and the translation type is **dnat-44**.

```
[edit services nat rule rule-dnat44 term t1]
user@host# set then port-forwarding-mappings map1 translation-type dnat-44
```

8. Go to the **[edit services nat port-forwarding map1]** hierarchy level.

```
[edit services nat]
user@host# edit port-forwarding map1
```

9. Configure the mapping for port forwarding.

```
[edit port-forwarding map1]
user@host# set destined-port port-id
user@host# set translated-port port-id
```

In the following example, the destination port is **45** and the translated port is **23**.

```
[edit port-forwarding map1]
```

```
user@host# set destined-port 23
user@host# set translated-port 45
```

NOTE:

- Multiple port mappings are supported with port forwarding. Up to 32 port maps can be configured for port forwarding.
- The destination port should not overlap the port range configured for NAT.

10. Verify the configuration by using the **show** command at the **[edit services nat]** hierarchy level.

```
[edit services]
user@host# show
nat {
 pool dest-pool {
     address 4.1.1.2/32;
 }
 rule rule-dnat44 {
     match-direction input;
     term t1 {
  from {
      destination-address {
          20.20.20.20/32;
      }
      destination-port {
          range low 20 high 50;
      }
  }
  then {
      port-forwarding-mappings map1;
      translated {
           destination-pool dest-pool;
           translation-type {
               dnat-44;
             }
          }
     }
     }
 }
 port-forwarding map1 {
     destined-port 45;
   translated-port 23;
     }
}
```

NOTE:

- A similar configuration is possible with twice NAT for IPv4. See "Example: Configuring Port Forwarding with Twice NAT" on page 60.
- Port forwarding and stateful firewall can be configured together. Stateful firewall has precedence over port forwarding.

Related Documentation
- Example: Configuring Static Destination Address Translation on page 80

Configuring Port Forwarding Without Destination Address Translation

Starting with Junos OS Release 12.1, you can configure port forwarding without translating a destination address.

To configure port forwarding without destination address translation in IPv4 networks:

1. In configuration mode, go to the **[edit services nat]** hierarchy level.

   ```
   [edit]
   user@host# edit services nat
   ```

2. Configure the rule, match direction, term, and destination address.

   ```
   [edit services nat]
   user@host# set rule rule-name match-direction match-direction term term-name
   ```

   In the following example, the name of the rule is **rule-port-forwarding**, the match direction is **input**, and the name of the term is **t1**.

   ```
   [edit services nat]
   user@host# set rule rule-port-forwarding match-direction input term t1
   ```

3. Go to the **[edit services nat rule rule-port-forwarding term t1]** hierarchy level.

   ```
   [edit services nat]
   user@host# edit rule rule-port-forwarding term t1
   ```

4. Specify that there is no address translation for this rule.

   ```
   [edit services nat rule rule-port-forwarding term t1]
   user@host# set then no-translation
   ```

5. Configure the mapping for port forwarding and the translation type.

   ```
   [edit services nat rule rule-port-forwarding term t1]
   user@host# set then port-forwarding-mappings  map-name
   ```

   In the following example, the port forwarding map name is **map1**.

   ```
   [edit services nat rule rule-port-forwarding term t1]
   user@host# set then port-forwarding-mappings map1
   ```

6. Go to the **[edit services nat port-forwarding map1]** hierarchy level.

   ```
   [edit services nat]
   user@host# edit port-forwarding map1
   ```

7. Configure the mapping for port forwarding.

   ```
   [edit port-forwarding map1]
   user@host# set destined-port port-id
   user@host# set translated-port port-id
   ```

   In the following example, the destination port is **45** and the translated port is **23**.

   ```
   [edit port-forwarding map1]
   user@host# set destined-port 23
   user@host# set translated-port 45
   ```

NOTE:

- Multiple port mappings are supported with port forwarding. Up to 32 port maps can be configured for port forwarding.
- The destination port should not overlap the port range configured for NAPT.

8. Verify the configuration by using the **show** command at the **[edit services nat]** hierarchy level.

```
[edit services]
user@host# show
nat {
 rule rule-port-forwarding {
     match-direction input;
     term t1 {
  then {
      port-forwarding-mappings map1;
      no-translation          }
     }
     }
 }
 port-forwarding map1 {
     destined-port 45;
   translated-port 23;
     }
}
```

NOTE: Port forwarding and stateful firewall can be configured together. Stateful firewall has precedence over port forwarding.

Configuring Secured Port Block Allocation

To configure secured port block allocation:

1. At the **[edit services nat pool *poolname*]** hierarchy level, create a pool.

   user@host# **edit services nat** pool ***poolname***

   For example:

   user@host# **edit services nat pool pba-pool1**

2. Define the range of addresses to be translated, specifying the upper and lower limits of the range or an address prefix that describes the range.

   [edit services nat pool pba-pool1]
   user@host# **set** address-range **low** ***address*** **high** ***address***

   Or

   user@host# **set** address ***address-prefix***

   For example:

```
[edit services nat pool pba-pool1]
user@host# set address 203.0.113.0/24
```

3. Define the range of ports to be used in the translation, or use automatic port assignment by the Junos OS. You can optionally specify random assignment of ports (sequential assignment is the default).

   ```
   [edit services nat pool pba-pool1]
   user@host# set port range low address high address random
   ```

   Or

   ```
   user@host# set port automatic random-allocation
   ```

   For example:

   ```
   [edit services nat pool pba-pool1]
   user@host# set port range low 256 high 511 random
   ```

   Or

   ```
   [edit services nat pool pba-pool1]
   user@host# set port automatic random-allocation
   ```

   NOTE: When you configure a port range, the range should be a multiple of the port block-size value (see Step 4). When the NAT pool port range is *not* a multiple of the port block-size value, the number of ports or port-blocks that are effectively available for use is less than the configured number of ports and port-blocks. The port block allocation mechanism uses ports in the range 0 through 1023 of a NAT address.

   When you configure automatic assignment of ports, the available port range for allocation is 1024 through 65535. Automatic allocation can result in no ports being available for use. Use the show services nat pool command on the Routing Engine after you configure the port block allocation method to determine the number of ports and port blocks available for allocation to users.

4. Configure secured port block allocation. Specify **active-block-timeout**, **block-size**, and **max-blocks-per-address**, or accept the default values for those options.

   ```
   [edit services nat pool pba-pool1]
   user@host# set secured-port-block-allocation active-block-timeout
     active-block-timeout block-size block-size max-blocks-per-address
     max-blocks-per-address
   ```

   For example:

   ```
   [edit services nat pool pba-pool1]
   user@host# set secured-port-block allocation active-block-timeout 120 block-size
     256  max-blocks-per-address 12
   ```

NOTE: You must reboot the services PIC whenever you:

- Change the block-size.
- Change the port range.
- Add or delete prefixes.
- Modify the from hierarchy in the NAT rule.

Related Documentation

- Configuring Addresses and Ports for Use in NAT Rules on page 21

Configuring Deterministic Port Block Allocation

To configure deterministic port block allocation:

1. At to the **[edit services nat pool *poolname*]** hierarchy level, create a pool.

   user@host# edit services nat pool *poolname*

   For example:

   user@host# edit services nat pool pba-pool2

2. Define the range of addresses to be translated, specifying the upper and lower limits of the range or an address prefix that describes the range.

   [edit services nat pool pba-pool1]
   user@host# set address-range low *address* high *address*

   Or

   user@host# set address *address-prefix*

   For example:

   [edit services nat pool pba-pool1]
   user@host# set address-range low 32.32.32.1 high 32.32.32.253

3. Specify automatic port assignment by the Junos OS.

   [edit services nat pool pba-pool1]
   user@host# set port automatic

4. Configure deterministic port block allocation. Specify **block-size** or accept the default value of 256.

   [edit services nat pool pba-pool1]
   user@host# set deterministic-port-block-allocation blocksize *block-size*

   For example:

   [edit services nat pool pba-pool1]
   user@host# set deterministic-port-block-allocation block-size *256*

NOTE: When you use deterministic port block allocation, you must reboot the services PIC if you:

- Change the block-size option.
- Change the port max-blocks-per-address option.

Related Documentation

- Configuring Addresses and Ports for Use in NAT Rules on page 21

CHAPTER 5

NAT Rules Examples

- Example: Configuring Static Source Translation in an IPv4 Network on page 76
- Example: Configuring Static Source Translation in an IPv6 Network on page 76
- Example: Configuring Static Source Translation with Multiple Prefixes and Address Ranges on page 77
- Example: Configuring Dynamic Source Address and Port Translation (NAPT) for an IPv4 Network on page 78
- Example: Configuring Dynamic Source Translation for an IPv4 Network on page 78
- Example: Configuring Dynamic Address-Only Source Translation on page 79
- Example: Configuring Dynamic Address-Only Source Translation in an IPv4 Network on page 79
- Example: Configuring Static Destination Address Translation on page 80
- Example: Configuring Dynamic Source Address and Static Destination Address Translation (IPv6 to IPV4) on page 81
- Example: Configuring NAT in Mixed IPv4 and IPv6 Networks on page 81
- Example: Configuring the Translation Type Between IPv6 and IPv4 Networks on page 84
- Example: Configuring Source Dynamic and Destination Static Translation on page 86
- Example: Configuring NAT-PT on page 86
- Example: Configuring an Oversubscribed Pool with Fallback to NAPT on page 99
- Example: Configuring an Oversubscribed Pool with No Fallback on page 99
- Example: Assigning Addresses from a Dynamic Pool for Static Use on page 100
- Example: Configuring NAT Rules Without Defining a Pool on page 101
- Example: Preventing Translation of Specific Addresses on page 101
- Example: Configuring NAT for Multicast Traffic on page 102
- Example: Configuring Twice NAT on page 106
- Example: Configuring Port Forwarding with Twice NAT on page 106
- Example: NAT 44 CGN Configurations on page 108
- Example: NAT Between VRFs Configuration on page 111
- Example: Configuring Stateful NAT64 for Handling IPv4 Address Depletion on page 114

Example: Configuring Static Source Translation in an IPv4 Network

The following configuration sets up one-to-one mapping between a private subnet and a public subnet.

```
[edit]
user@host# show services
service-set s1 {
    nat-rules rule-basic-nat44;
    interface-service {
        service-interface ms-1/2/0;
    }
}
nat {
    pool src_pool {
        address 10.10.10.2/32;
    }
    rule rule-basic-nat44 {
        match-direction input;
        term t1 {
            from {
                source-address {
                    3.1.1.2/32;
                }
            }
            then {
                translated {
                    source-pool src_pool;
                    translation-type {
                        basic-nat44;
                    }
                }
            }
        }
    }
}
adaptive-services-pics {
    traceoptions {
        flag all;
    }
}
```

Example: Configuring Static Source Translation in an IPv6 Network

The following example configures the translation type as **basic-nat66**.

```
[edit]
user@host# show services
service-set s1 {
    nat-rules rule-basic-nat66;
    interface-service {
        service-interface sp-1/2/0;
    }
}
nat {
    pool src_pool {
```

```
        address 10.10.10.2/32;
    }
    rule rule-basic-nat66 {
        match-direction input;
        term t1 {
            from {
                source-address {
                    10:10:10::0/96;
                }
            }
            then {
                translated {
                    source-pool src_pool;
                    translation-type {
                        basic-nat66;
                    }
                }
            }
        }
    }
}
adaptive-services-pics {
    traceoptions {
        flag all;
    }
}
```

Example: Configuring Static Source Translation with Multiple Prefixes and Address Ranges

The following configuration creates a static pool with an address prefix and an address range and uses static source NAT translation.

```
[edit services nat]
pool p1 {
  address 30.30.30.252/30;
  address-range low 20.20.20.1 high 20.20.20.2;
}
rule r1 {
  match-direction input;
  term {
    from {
      source-address {
        10.10.10.252/30;
      }
    }
    then {
      translated {
        source-pool p1;
        translation-type basic-nat44;
      }
    }
  }
}
```

Example: Configuring Dynamic Source Address and Port Translation (NAPT) for an IPv4 Network

The following example configures dynamic source (address and port) translation, or NAPT.

```
[edit services nat]
pool public {
  address-range low 192.16.2.1 high 192.16.2.32;
  port automatic;
}
rule Private-Public {
  match-direction input;
  term Translate {
    then {
      translated {
        source-pool public;
        translation-type napt-44;
      }
    }
  }
}
```

NOTE: The only difference between the configurations for dynamic address-only source translation and NAPT is the inclusion of the **port** statement for NAPT.

Example: Configuring Dynamic Source Translation for an IPv4 Network

The following example configures the translation type as **napt-44**.

```
[edit services]
    user@host# show
    service-set s1 {
    nat-rules rule-napt-44;
    interface-service {
        service-interface ms-0/1/0;
    }
}
nat {
    pool napt-pool {
        address 10.10.10.0/32;
        port {
            automatic;
        }
    }
    rule rule-napt-44 {
        match-direction input;
        term t1 {
            then {
                translated {
                    source-pool napt-pool;
                    translation-type {
                        napt-44;
```

```
                    }
                }
            }
        }
    }
}
adaptive-services-pics {
    traceoptions {
        flag all;
    }
}
```

Example: Configuring Dynamic Address-Only Source Translation

The following example configures dynamic address-only source translation.

```
[edit services nat]
pool public {
  address-range low 192.16.2.1 high 192.16.2.32;
}
rule Private-Public {
  match-direction input;
  term Translate {
    then {
      translated {
        source-pool public;
        translation-type dynamic-nat44 ;
      }
    }
  }
}
```

Example: Configuring Dynamic Address-Only Source Translation in an IPv4 Network

The following example configures the translation type as **dynamic-nat44**.

```
[edit services]
user@host# show
service-set s1 {
    nat-rules rule-dynamic-nat44;
    interface-service {
        service-interface ms-0/1/0;
    }
}
 nat {
    pool source-dynamic-pool {
        address 10.1.1.0/24;
    }
    rule rule-dynamic-nat44 {
        match-direction input;
        term t1 {
            from {
                source-address {
                    3.1.1.0/24;
                }
            }
            then {
```

```
                translated {
                    destination-pool source-dynamic-pool;
                    translation-type {
                        dynamic-nat44;
                    }
                }
            }
        }
    }
}
 adaptive-services-pics {
    traceoptions {
        flag all;
    }
  }
```

Example: Configuring Static Destination Address Translation

The following example configures the translation type as **dnat-44**.

```
[edit services]
user@host# show
service-set s1 {
    nat-rules rule-dnat44;
    interface-service {
        service-interface ms-0/1/0;
    }
}
 nat {
    pool dest-pool {
        address 4.1.1.2/32;
    }
    rule rule-dnat44 {
        match-direction input;
        term t1 {
            from {
                destination-address {
                    20.20.20.20/32;
                }
            }
            then {
                translated {
                    destination-pool dest-pool;
                    translation-type {
                        dnat-44;
                    }
                }
            }
        }
    }
}
  adaptive-services-pics {
    traceoptions {
        flag all;
    }
  }
```

Example: Configuring Dynamic Source Address and Static Destination Address Translation (IPv6 to IPV4)

The following example configures dynamic source address (IPv6-to-IPv4) and static destination address (IPv6-to-IPv4) translation:

```
[edit services]
user@host# show
nat {
    pool src-pool-nat64 {
        address 203.0.113.0/24;
        port {
            automatic;
        }
    }
    rule stateful-nat64 {
        match-direction input;
        term t1 {
            from {
                source-address {
                    2001:db8::0/96;
                }
                destination-address {
                    64:ff9b::/96;
                }
            }
            then {
                translated {
                    source-pool src-pool-nat64;
                    destination-prefix 64:ff9b::/96;
                    translation-type {
                        stateful-nat64;
                    }
                }
            }
        }
    }
}
service-set sset-nat64 {
    nat-options {
        stateful-nat64 {
            clear-dont-fragment-bit;
        }
    }
    service-set-options;
    nat-rules stateful-nat64;
    interface-service {
        service-interface ms-0/1/0;
    }
}
```

Example: Configuring NAT in Mixed IPv4 and IPv6 Networks

The following example shows NAT configuration in a network that combines IPv4 and IPv6 addressing.

interfaces {

```
sp-2/0/0 {
    traceoptions {
        flag all;
    }
    services-options {
        syslog {
            host local {
                services any;
                log-prefix IPV6-SS;
            }
        }
        inactivity-timeout 200;
    }
    unit 0 {
        family inet;
        family inet6;
    }
    unit 1 {
        family inet;
        family inet6;
    }
    unit 1001 {
        family inet;
        family inet6;
    }
}
so-2/1/0 {
    description "services-art1 201/1";
    encapsulation ppp;
    sonet-options {
        fcs 32;
    }
    unit 0 {
        family inet {
            address 192.168.1.1/30;
        }
        family inet6 {
            service {
                input {
                    service-set ss-ipv6;
                }
                output {
                    service-set ss-ipv6;
                }
            }
            address 3ffe::1:1/64;
        }
    }
}
so-2/1/3 {
    description "services-art1 201/2";
    encapsulation ppp;
    sonet-options {
        fcs 32;
    }
    unit 0 {
```

```
            family inet {
                address 192.168.1.1/30;
            }
            family inet6 {
                address 4ffe::1:1/64;
            }
        }
    }
}
    routing-options {
        rib inet6.0 {
            static {
                route 5ffe::0:0/64 next-hop 3ffe::1:1;
                route 6ffe::0:0/64 next-hop 4ffe::1:1;
                route 192::168:1:0/112 next-hop 4ffe::1:2;
            }
        }
    }
    services {
        service-set ss-ipv6 {
            stateful-firewall-rules test-ipv6;
            nat-rules src-nat;
            nat-rules src-nat-v6;
            interface-service {
                service-interface sp-2/0/0;
            }
        }
        stateful-firewall {
            rule test-ipv6 {
                match-direction input-output;
                term 1 {
                    from {
                        source-address {
                            any-unicast;
                        }
                    }
                    then {
                        accept;
                    }
                }
            }
        }
        nat {
            pool dst_pool {
                address 192.168.1.2/32;
            }
            pool src_pool {
                address 192.168.1.0/27;
                port automatic;
            }
            pool dst_pool_v6 {
                address 192::168:1:2/128;
            }
            pool src_pool_v6 {
                address 192::168:1:2/100;
            }
```

```
        rule src-nat {
            match-direction input;
            term t1 {
                from {
                    source-address {
                        3ffe::0:0/96;
                    }
                    destination-address {
                        4ffe::1:2/128;
                    }
                }
                then {
                    translated {
                        source-pool src_pool;
                        destination-pool dst_pool;
                        translation-type {
                            source dynamic;
                            destination static;
                        }
                    }
                }
            }
        }
        rule src-nat-v6 {
            match-direction input;
            term t1 {
                from {
                    source-address {
                        5ffe::0:0/96;
                    }
                    destination-address {
                        6ffe::1:2/128;
                    }
                }
                then {
                    translated {
                        source-pool src_pool_v6;
                        destination-pool dst_pool_v6;
                        translation-type {
                            source static;
                            destination static;
                        }
                    }
                }
            }
        }
    }
}
```

Example: Configuring the Translation Type Between IPv6 and IPv4 Networks

The following example configures the translation type as **basic-nat-pt**.

```
[edit]
user@host# show services
```

```
service-set ss_dns {
    nat-rules rule-basic-nat-pt;
    interface-service {
        service-interface sp-1/2/0;
    }
}
nat {
    pool p1 {
        address 10.10.10.2/32;
    }
    pool src_pool0 {
        address 20.1.1.1/32;
    }
    pool dst_pool0 {
        address 50.1.1.2/32;
    }
    rule rule-basic-nat-pt {
        match-direction input;
        term t1 {
            from {
                source-address {
                    2000::2/128;
                }
                destination-address {
                    4000::2/128;
                }
                applications dns_alg;
            }
            then {
                translated {
                    source-pool src_pool0;
                    destination-pool dst_pool0;
                    dns-alg-prefix 10:10:10::0/96;
                    translation-type {
                        basic-nat-pt;
                    }
                }
            }
        }
        term t2 {
            from {
                source-address {
                    2000::2/128;
                }
                destination-address {
                    10:10:10::0/96;
                }
            }
            then {
                translated {
                    source-prefix 19.19.19.1/32;
                    translation-type {
                        basic-nat-pt;
                    }
                }
            }
        }
    }
}
adaptive-services-pics {
    traceoptions {
```

```
        flag all;
    }
}
```

Example: Configuring Source Dynamic and Destination Static Translation

In the following configuration, **term1** configures source address translation for traffic from any private address to any public address. The translation is applied for all services. **term2** performs destination address translation for Hypertext Transfer Protocol (HTTP) traffic from any public address to the server's virtual IP address. The virtual server IP address is translated to an internal IP address.

```
[edit services nat]
rule my-nat-rule {
  match-direction input;
  term my-term1 {
    from {
      source-address private;
      destination-address public;
    }
    then {
      translated {
        source-pool my-pool; # pick address from a pool
        translation-type napt-44; # dynamic NAT with port translation
      }
    }
  }
  term my-term2 {
    from {
      destination-address 192.168.137.3; # my server's virtual address
      application http;
    }
    then {
      translated {
        destination-pool nat-pool-name;
        translation-type dnat-44; # static destination NAT
      }
    }
  }
}
```

Example: Configuring NAT-PT

A Domain Name System application-level gateway (DNS ALG) is used with Network Address Translation-Protocol Translation (NAT-PT) to facilitate name-to-address mapping. You can configure the DNS ALG to map addresses returned in the DNS response to an IPv6 address.

When you configure NAT-PT with DNS ALG support, you must configure two NAT rules or one rule with two terms. In this example, you configure two rules. The first NAT rule ensures that the DNS query and response packets are translated correctly. For this rule to work, you must configure a DNS ALG application and reference it in the rule. The second

rule is required to ensure that NAT sessions are destined to the address mapped by the DNS ALG.

Then, you must configure a service set, and then apply the service set to the interfaces.

This example describes how to configure NAT-PAT with DNS ALG:

- Requirements on page 87
- Overview and Topology on page 87
- Configuration of NAT-PT with DNS ALGs on page 88

Requirements

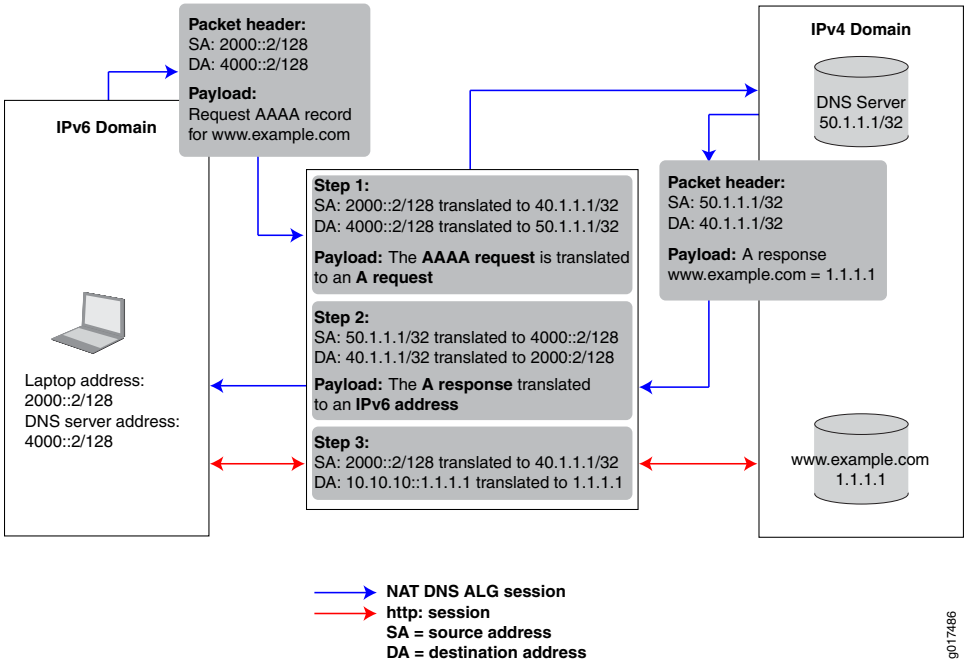
This example uses the following hardware and software components:

- Junos OS Release 11.2
- A multiservices interface (**ms-**)

Overview and Topology

The following scenario shows the process of NAT-PT with DNS ALG when a laptop in an IPv6-only domain requests access to a server in an IPv4-only domain.

Figure 7: Configuring DNS ALGs with NAT-PT Network Topology



The Juniper Networks router in the center of the illustration performs address translation in two steps. When the laptop requests a session with the **www.example.com** server that is in an IPv4-only domain, the Juniper Networks router performs the following:

- Translates the IPv6 laptop and DNS server addresses into IPv4 addresses.

- Translates the AAAA request from the laptop into an A request so that the DNS server can provide the IPv4 address.

When the DNS server responds with the A request, the Juniper Networks router performs the following:

- Translates the IPv4 DNS server address back into an IPv6 address.
- Translates the A request back into a AAAA request so that the laptop now has the 96-bit IPv6 address of the **www.example.com** server.

After the laptop receives the IPv6 version of the **www.example.com** server address, the laptop initiates a second session using the 96-bit IPv6 address to access that server. The Juniper Networks router performs the following:

- Translates the laptop IPv4 address directly into its IPv4 address.
- Translates the 96-bit IPv6 **www.example.com** server address into its IPv4 address.

Configuration of NAT-PT with DNS ALGs

To configure NAT-PT with DNS ALG , perform the following tasks:

- Configuring the Application-Level Gateway on page 88
- Configuring the NAT Pools on page 89
- Configuring the DNS Server Session: First NAT Rule on page 90
- Configuring the HTTP Session: Second NAT Rule on page 93
- Configuring the Service Set on page 95
- Configuring the Stateful Firewall Rule on page 96
- Configuring Interfaces on page 97

Configuring the Application-Level Gateway

Step-by-Step Procedure

Configure the DNS application as the ALG to which the DNS traffic is destined. The DNS application protocol closes the DNS flow as soon as the DNS response is received. When you configure the DNS application protocol, you must specify the UDP protocol as the network protocol to match in the application definition.

To configure the DNS application:

1. In configuration mode, go to the **[edit applications]** hierarchy level:

   ```
   user@host# edit applications
   ```

2. Define the application name and specify the application protocol to use in match conditions in the first NAT rule.

   ```
   [edit applications]
   user@host#  set application application-name application-protocol protocol-name
   ```

   For example:

   ```
   [edit applications]
   user@host#  set application dns_alg application-protocol dns
   ```

3. Specify the protocol to match, in this case UDP.

   ```
   [edit applications]
   user@host#  set application application-name protocol type
   ```

   For example:

   ```
   [edit applications]
   user@host# set application dns_alg protocol udp
   ```

4. Define the UDP destination port for additional packet matching, in this case the domain port.

   ```
   [edit applications]
   user@host#  set application application-name destination-port value
   ```

   For example:

   ```
   [edit applications]
   user@host# set application dns_alg destination-port 53
   ```

Results

```
[edit applications]
user@host# show
application dns_alg {
    application-protocol dns;
    protocol udp;
    destination-port 53;
}
```

Configuring the NAT Pools

Step-by-Step Procedure

In this configuration, you configure two pools that define the addresses (or prefixes) used for NAT. These pools define the IPv4 addresses that are translated into IPv6 addresses. The first pool includes the IPv4 address of the source. The second pool defines the IPv4 address of the DNS server. To configure NAT pools:

1. In configuration mode, go to the **[edit services nat]** hierarchy level.

   ```
   user@host# edit services nat
   ```

2. Specify the name of the first pool and the IPv4 source address (laptop).

   ```
   [edit services nat]
   user@host# set pool nat-pool-name address ip-prefix
   ```

   For example:

   ```
   [edit services nat]
   user@host# set pool pool1 address 40.1.1.1/32
   ```

3. Specify the name of the second pool and the IPv4 address of the DNS server.

   ```
   [edit services nat]
   user@host# set pool nat-pool-name address ip-prefix
   ```

   For example:

   ```
   [edit services nat]
   user@host# set pool pool2 address 50.1.1.1/32
   ```

Results The following sample output shows the configuration of NAT pools:

```
[edit services nat]
user@host# show
pool pool1 {
    address 40.1.1.1/32;
}
pool pool2 {
    address 50.1.1.1/32;
}
```

Configuring the DNS Server Session: First NAT Rule

Step-by-Step Procedure The first NAT rule is applied to DNS traffic going to the DNS server. This rule ensures that the DNS query and response packets are translated correctly. For this rule to work, you must configure a DNS ALG application and reference it in the rule. The DNS application was configured in "Configuring the DNS ALG Application" on page 53. In addition, you must specify the direction in which traffic is matched, the source address of the laptop, the destination address of the DNS server, and the actions to take when the match conditions are met.

To configure the first NAT rule:

1. In configuration mode, go to the **[edit services nat]** hierarchy level.

   user@host# **edit services nat**

2. Specify the name of the NAT rule.

   [edit services nat]
   user@host# **edit rule** ***rule-name***

   For example:

   [edit services nat]
   user@host# **edit rule rule1**

3. Specify the name of the NAT term.

   [edit services nat rule *rule-name*]
   user@host# **edit term** ***term-name***

   For example:

   [edit services nat rule rule1]
   user@host# **edit term term1**

4. Define the match conditions for this rule.

   - Specify the IPv6 source address of the device (laptop) attempting to access an IPv4 address.

     [edit services nat rule *rule-name* term *term-name*]
     user@host# **set from source-address** ***source-address***

     For example:

     [edit services nat rule rule1 term term1]
     user@host# **set from source-address 2000::2/128**

- Specify the IPv6 destination address of the DNS server.

  [edit services nat rule *rule-name* term *term-name*]
  user@host# **set from destination-address** ***prefix***

  For example:

  [edit services nat rule rule1 term term1]
  user@host# **set from destination-address 4000::2/128**

- Reference the DNS application to which the DNS traffic destined for port 53 is applied.

  [edit services nat rule rule1 term term1]
  user@host# **set from applications** ***application-name***

  In this example, the application name configured in the *Configuring the DNS Application* step is **dns_alg**:

  [edit services nat rule rule1 term term1]
  user@host# **set from applications dns_alg**

5. Define the actions to take when the match conditions are met. The source and destination pools you configured in "Configuring the NAT Pools" on page 89are applied here.

   - Apply the NAT pool configured for source translation.

     [edit services nat rule *rule-name* term *term-name*]
     user@host# **set then translated source-pool** ***nat-pool-name***

     For example:

     [edit services nat rule rule1 term term1]
     user@host# **set then translated source-pool pool1**

   - Apply the NAT pool configured for destination translation.

     [edit services nat rule *rule-name* term *term-name*]
     user@host# **set then translated destination-pool** ***nat-pool-name***

     For example:

     [edit services nat rule rule1 term term1]
     user@host# **set then translated source-pool pool2**

6. Define the DNS ALG 96-bit prefix for IPv4-to-IPv6 address mapping.

   [edit services nat rule *rule-name* term *term-name*]
   user@host# **set then translated dns-alg-prefix** ***dns-alg-prefix***

   For example:

   [edit services nat rule rule1 term term1]
   user@host# **set then translated dns-alg-prefix 10:10:10::0/96**

7. Specify the type of NAT used for source and destination traffic.

   [edit services nat rule *rule-name* term *term-name*]
   user@host# **set then translated translation-type basic-nat-pt**

   For example:

```
[edit services nat rule rule1 term term1]
user@host# set then translated translation-type basic-nat-pt
```

NOTE: In this example, since NAT is achieved using address-only translation, the **basic-nat-pt** translation type is used. To achieve NAT using address and port translation (NAPT), use the **napt-pt** translation type.

8. Specify the direction in which to match traffic that meets the rule conditions.

   ```
   [edit services nat rule rule-name]
   user@host# set match-direction (input | output)
   ```

   For example:

   ```
   [edit services nat rule rule1]
   user@host# set match-direction input
   ```

9. Configure system logging to record information from the services interface to the /var/log directory.

   ```
   [edit services nat rule rule-name term term-name]
   user@host# set then syslog
   ```

   For example:

   ```
   [edit services nat rule rule1 term term1]
   user@host# set then syslog
   ```

Results The following sample output shows the configuration of the first NAT rule that goes to the DNS server.

```
[edit services nat]
user@host# show
rule rule1 {
    match-direction input;
    term term1 {
        from {
            source-address {
                2000::2/128;
            }
            destination-address {
                4000::2/128;
            }
            applications dns_alg;
        }
        then {
            translated {
                source-pool pool1;
                destination-pool pool2;
                dns-alg-prefix 10:10:10::0/96;
                translation-type {
                    basic-nat-pt;

                }
            }
            syslog;
```

```
        }
    }
}
```

Configuring the HTTP Session: Second NAT Rule

Step-by-Step Procedure

The second NAT rule is applied to destination traffic going to the IPv4 server **www.example.com**). This rule ensures that NAT sessions are destined to the address mapped by the DNS ALG. For this rule to work, you must configure the DNS ALG address map that correlates the DNS query or response processing done by the first rule with the actual data sessions processed by the second rule. In addition, you must specify the direction in which traffic is matched: the IPv4 address for the IPv6 source address (laptop), the 96-bit prefix to prepend to the IPv4 destination address (www.example.com), and the translation type.

To configure the second NAT rule:

1. In configuration mode, go to the following hierarchy level.

   user@host# edit services nat

2. Specify the name of the NAT rule and term.

   [edit services nat]
   user@host# edit rule *rule-name* term *term-name*

   For example:

   [edit services nat]
   user@host# edit rule rule2 term term1

3. Define the match conditions for this rule:

   - Specify the IPv6 address of the device attempting to access the IPv4 server.

     [edit services nat rule *rule-name* term *term-name*]
     user@host# set from source-address *source-address*

     For example:

     [edit services nat rule rule2 term term1]
     user@host# set from source-address 2000::2/128

   - Specify the 96-bit IPv6 prefix to prepend to the IPv4 server address.

     [edit services nat rule *rule-name* term *term-name*]
     user@host# set from destination-address *prefix*

     For example:

     [edit services nat rule rule2 term term1]
     user@host# set from destination-address 10:10:10::c0a8:108/128

4. Define the actions to take when the match conditions are met.

   - Specify the prefix for the translation of the IPv6 source address.

     [edit services nat rule *rule-name* term *term-name*]
     user@host# set then translated source-prefix *source-prefix*

     For example:

```
[edit services nat rule rule2 term term1]
user@host# set then translated source-prefix 19.19.19.1/32
```

5. Specify the type of NAT used for source and destination traffic.

```
[edit services nat rule rule-name term term-name]
user@host# set then translated translation-type basic-nat-pt
```

For example:

```
[edit services nat rule rule2 term term1]
user@host# set then translated translation-type basic-nat-pt
```

NOTE: In this example, since NAT is achieved using address-only translation, the basic-nat-pt translation type is used. To achieve NAT using address and port translation (NAPT), you must use the napt-pt translation type.

6. Specify the direction in which to match traffic that meets the conditions in the rule.

```
[edit services nat rule rule-name]
user@host# set match-direction (input | output)
```

For example:

```
[edit services nat rule rule2]
user@host# set match-direction input
```

Results The following sample output shows the configuration of the second NAT rule:

```
[edit services nat]
user@host# show
rule rule2 {
    match-direction input;
    term term1 {
        from {
            source-address {
                2000::2/128;
            }
            destination-address {
                10:10:10::c0a8:108/128;
            }
        }
        then {
            translated {
                source-prefix 19.19.19.1/32;
                translation-type {
                    basic-nat-pt;
                }
            }
        }
    }
}
```

Configuring the Service Set

Step-by-Step Procedure

This service set is an interface service set used as an action modifier across the entire services (**ms-**) interface. Stateful firewall and NAT rule sets are applied to traffic processed by the services interface.

To configure the service set:

1. In configuration mode, go to the **[edit services]** hierarchy level.

   ```
   user@host# edit services
   ```

2. Define a service set.

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

   For example:

   ```
   [edit services]
   user@host# edit service-set ss
   ```

3. Specify properties that control how system log messages are generated for the service set.

   ```
   [edit services service-set ss]
   user@host# set syslog host local services severity-level
   ```

   The example below includes all severity levels.

   ```
   [edit services service-set ss]
   user@host# set syslog host local services any
   ```

4. Specify the stateful firewall rule included in this service set.

   ```
   [edit services service-set ss]
   user@host# set stateful-firewall-rules rule1 severity-level
   ```

   The example below references the stateful firewall rule defined in "Configuring the Stateful Firewall Rule" on page 96.

   ```
   [edit services service-set ss]
   user@host# set stateful-firewall-rules rule1
   ```

5. Define the NAT rules included in this service set.

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

   The example below references the two rules defined in this configuration example.

   ```
   [edit services service-set ss
   user@host# set nat-rules rule1
   user@host# set nat-rules rule2
   ```

6. Configure an adaptive services interface on which the service is to be performed.

   ```
   [edit services service-set ss]
   user@host# set interface-service service-interface interface-name
   ```

   For example:

   ```
   [edit services service-set ss
   ```

user@host# **interface-service service-interface ms-2/0/0**

Only the device name is needed, because the router software manages logical unit numbers automatically. The services interface must be an adaptive services interface for which you have configured **unit 0 family inet** at the **[edit interfaces *interface-name*]** hierarchy level in the "Configuring Interfaces" on page 97 step.

Results The following sample output shows the configuration of the service set:

```
[edit services]
user@host# show
service-set ss {
    syslog {
        host local {
            services any;
        }
    }
    stateful-firewall-rules rule1;
    nat-rules rule1;
    nat-rules rule2;
    interface-service {
        service-interface ms-2/0/0;
    }
}
```

Configuring the Stateful Firewall Rule

Step-by-Step Procedure This example uses a stateful firewall to inspect packets for state information derived from past communications and other applications. The NAT-PT router checks the traffic flow matching the direction specified by the rule, in this case both input and output. When a packet is sent to the services (**ms-**) interface, direction information is carried along with it.

To configure the stateful firewall rule:

1. In configuration mode, go to the **[edit services stateful firewall]** hierarchy level.

   user@host# **edit services stateful firewall**

2. Specify the name of the stateful firewall rule.

   [edit services stateful-firewall]
   user@host# **edit rule *rule-name***

   For example:

   [edit services stateful-firewall]
   user@host# **edit rule rule1**

3. Specify the direction in which traffic is to be matched.

   [edit services stateful-firewall rule *rule-name*]
   user@host# **set match-direction (input | input-output | output)**

   For example:

   [edit services stateful-firewall rule rule1]
   user@host# **set match-direction *input-output***

4. Specify the name of the stateful firewall term.

[edit services stateful-firewall rule *rule-name*]
user@host# **edit term *term-name***

For example:

[edit services stateful-firewall rule rule1]
user@host# **edit term term1**

5. Define the terms that make up this rule.

[edit services stateful-firewall rule *rule-name* term *term-name*]
user@host# **set then accept**

For example:

[edit services stateful-firewall rule rule1 term term1]
user@host# **set then accept**

Results The following sample output shows the configuration of the services stateful firewall.

```
[edit services]
user@host# show
stateful-firewall {
    rule rule1 {
        match-direction input-output;
        term term1 {
            then {
                accept;
            }
        }
    }
}
```

Configuring Interfaces

Step-by-Step Procedure After you have defined the service set, you must apply services to one or more interfaces installed on the router. In this example, you configure one interface on which you apply the service set for input and output traffic. When you apply the service set to an interface, it automatically ensures that packets are directed to the services (**ms-**) interface.

To configure the interfaces:

1. In configuration mode, go to the **[edit interfaces]** hierarchy level.

user@host# **edit interfaces**

2. Configure the interface on which the service set is applied to automatically ensure that packets are directed to the services (**ms-**) interface.

- For IPv4 traffic, specify the IPv4 address.

[edit interfaces]
user@host# **set ge-1/0/9 unit 0 family inet address 30.1.1.1/24**

- Apply the service set defined in the "Configuring Interfaces" on page 97 step.

[edit interfaces]
user@host# **set ge-1/0/9 unit 0 family inet6 service input service-set ss**
user@host# **set ge-1/0/9 unit 0 family inet6 service output service-set ss**

- For IPv6 traffic, specify the IPv6 address.

  ```
  [edit interfaces]
  user@host# set ge-1/0/9 unit 0 family inet6 address 2000::1/64
  ```

3. Specify the interface properties for the services interface that performs the service.

   ```
   [edit interfaces]
   user@host# set ms-2/0/0 services-options syslog host local services any
   user@host# set ms-2/0/0 unit 0 family inet
   user@host# set ms-2/0/0 unit 0 family inet6
   ```

Results The following sample output shows the configuration of the interfaces for this example.

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

ge-1/0/9 {
    unit 0 {
        family inet {
            address 30.1.1.1/24;
        }
        family inet6 {
            service {
                input {
                    service-set ss;
                }
                output {
                    service-set ss;
                }
            }
            address 2000::1/64;
        }
    }
}

ms-2/0/0 {
    services-options {
        syslog {
            host local {
                services any;
            }
        }
    }
    unit 0 {
        family inet;
        family inet6;
    }
}
```

Related Documentation

- Network Address Translation Overview on page 7
- Configuring NAT-PT on page 58
- Configuring Service Sets to be Applied to Services Interfaces
- Example: Configuring the uKernel Service and the Services SDK on Two PICs
- dns-alg-prefix on page 133

- dns-alg-pool on page 132

Example: Configuring an Oversubscribed Pool with Fallback to NAPT

The following configuration shows dynamic address translation from a large prefix to a small pool, translating a **/24** subnet to a pool of 10 addresses. When the addresses in the source pool (**src-pool**) are exhausted, NAT is provided by the NAPT overload pool (**pat-pool**).

```
[edit services nat]
pool src-pool {
   address-range low 192.16.2.1 high 192.16.2.10;
}
pool pat-pool {
   address-range low 192.2.11 high 192.16.2.12;
   port automatic;
}
rule myrule {
   match-direction input;
   term myterm {
      from {
         source-address 10.150.1.0/24;
      }
      then {
         translated {
            source-pool src-pool;
            overload-pool pat-pool;
            translation-type napt-44;
         }
      }
   }
}
```

Example: Configuring an Oversubscribed Pool with No Fallback

The following configuration shows dynamic address translation from a large prefix to a small pool, translating a **/24** subnet to a pool of 10 addresses. Sessions from the first 10 host sessions are assigned an address from the pool on a first-come, first-served basis, and any additional requests are rejected. Each host with an assigned NAT can participate in multiple sessions.

```
[edit services nat]
pool my-pool {
   address-range low 10.10.10.1 high 10.10.10.10;
}
rule src-nat {
   match-direction input;
   term t1 {
      from {
         source-address 192.168.1.0/24;
      }
      then {
         translated {
```

```
            translation-type dynamic-nat44;
            source-pool my-pool;
          }
        }
      }
    }
```

Example: Assigning Addresses from a Dynamic Pool for Static Use

The following configuration statically assigns a subset of addresses that are configured as part of a dynamic pool (**dynamic-pool**) to two separate static pools (**static-pool** and **static-pool2**).

```
[edit services nat]
pool dynamic-pool {
  address 20.20.10.0/24;
}
pool static-pool {
  address-range low 20.20.10.10 high 10.20.10.12;
}
pool static-pool2 {
  address 20.20.10.15/32;
}
rule src-nat {
  match-direction input;
  term t1 {
    from {
      source-address 30.30.30.0/24;
    }
    then {
      translation-type dynamic-nat44;
      source-pool dynamic-pool;
    }
  }
  term t2 {
    from {
      source-address 10.10.10.2;
    }
    then {
      translation-type basic-nat44;
      source-pool static-pool;
    }
  }
  term t3 {
    from {
      source-address 10.10.10.10;
    }
    then {
      translation-type basic-nat44;
      source-pool static-pool2;
    }
  }
}
```

Example: Configuring NAT Rules Without Defining a Pool

The following configuration performs NAT using the source prefix **20.20.10.0/24** without defining a pool.

```
[edit services nat]
rule src-nat {
   match-direction input;
   term t1 {
      then {
         translation-type dynamic-nat44;
         source-prefix 20.20.10.0/24;
      }
   }
}
```

The following configuration performs NAT using the destination prefix **20.20.10.0/32** without defining a pool.

```
[edit services nat]
rule src-nat {
   match-direction input;
   term t1 {
      from {
         destination-address 10.10.10.10/32;
         then {
            translation-type dnat44;
            destination-prefix 20.20.10.0/24;
         }
      }
   }
}
```

Example: Preventing Translation of Specific Addresses

The following configuration specifies that NAT is not performed on incoming traffic from the source address **192.168.20.24/32**. Dynamic NAT is performed on all other incoming traffic.

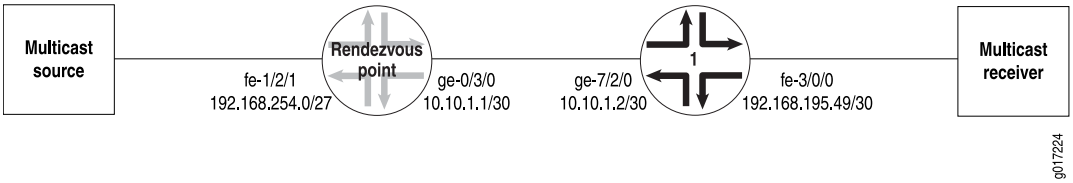
```
[edit services nat]
pool my-pool {
   address-range low 10.10.10.1 high 10.10.10.16;
   port-automatic;
}
rule src-nat {
   match-direction input;
   term t0 {
      from {
         source-address 192.168.20.24/32;
      }
      then {
         no-translation;
      }
   }
```

```
    term t1 {
      then {
        translated {
          translation-type dynamic-nat44;
          source-pool my-pool;
        }
      }
    }
  }
```

Example: Configuring NAT for Multicast Traffic

Figure 8 on page 102 illustrates the network setup for the following configuration, which allows IP multicast traffic to be sent to the Multiservices PIC.

Figure 8: Configuring NAT for Multicast Traffic



- Rendezvous Point Configuration on page 102
- Router 1 Configuration on page 105

Rendezvous Point Configuration

On the rendezvous point (RP), all incoming traffic from the multicast source at **192.168.254.0/27** is sent to the static NAT pool **mcast_pool**, where its source is translated to **20.20.20.0/27**. The service set **nat_ss** is a next-hop service set that allows IP multicast traffic to be sent to the Multiservices DPC or Multiservices PIC. The inside interface on the PIC is **ms-1/1/0.1** and the outside interface is **ms-1/1/0.2**.

```
[edit services]
nat {
  pool mcast_pool {
    address 20.20.20.0/27;
  }
  rule nat_rule_1 {
    match-direction input;
    term 1 {
      from {
        source-address 192.168.254.0/27;
      }
    }
    then {
      translated {
        source-pool mcast_pool;
        translation-type basic-nat44;
      }
      syslog;
    }
  }
```

```
}
service-set nat_ss {
    allow-multicast;
    nat-rules nat_rule_1;
    next-hop-service {
        inside-service-interface ms-1/1/0.1;
        outside-service-interface ms-1/1/0.2;
    }
}
```

The Gigabit Ethernet interface **ge-0/3/0** carries traffic out of the RP to Router 1. The multiservices interface **ms-1/1/0** has two logical interfaces: **unit 1** is the inside interface for next-hop services and **unit 2** is the outside interface for next-hop services. Multicast source traffic comes in on the Fast Ethernet interface **fe-1/2/1**, which has the firewall filter **fbf** applied to incoming traffic.

```
[edit interfaces]
ge-0/3/0 {
    unit 0 {
        family inet {
            address 10.10.1.1/30;
        }
    }
}
ms-1/1/0 {
    unit 0 {
        family inet;
    }
    unit 1 {
        family inet;
        service-domain inside;
    }
    unit 2 {
        family inet;
        service-domain outside;
    }
}
fe-1/2/1 {
    unit 0 {
        family inet {
            filter {
                input fbf;
            }
            address 192.168.254.27/27;
        }
    }
}
```

Multicast packets can only be directed to the Multiservices DPC or the Multiservices PIC using a next-hop service set. In the case of NAT, you must also configure a VRF. Therefore, the routing instance **stage** is created as a "dummy" forwarding instance. To direct incoming packets to **stage**, you configure filter-based forwarding through a firewall filter called **fbf**, which is applied to the incoming interface **fe-1/2/1**. A lookup is performed in **stage.inet.0**, which has a multicast static route that is installed with the next hop pointing to the PIC's inside interface. All multicast traffic matching this route is sent to the PIC.

```
[edit firewall]
filter fbf {
   term 1 {
      then {
         routing-instance stage;
      }
   }
}
```

The routing instance **stage** forwards IP multicast traffic to the inside interface **ms-1/1/0.1** on the Multiservices DPC or Multiservices PIC:

```
[edit]
routing-instances stage {
   instance-type forwarding;
   routing-options {
      static {
         route 224.0.0.0/4 next-hop ms-1/1/0.1;
      }
   }
}
```

You enable OSPF and Protocol Independent Multicast (PIM) on the Fast Ethernet and Gigabit Ethernet logical interfaces over which IP multicast traffic enters and leaves the RP. You also enable PIM on the outside interface (**ms-1/1/0.2**) of the next-hop service set.

```
[edit protocols]
ospf {
   area 0.0.0.0 {
      interface fe-1/2/1.0 {
         passive;
      }
      interface lo0.0;
      interface ge-0/3/0.0;
   }
}
pim {
   rp {
      local {
         address 10.255.14.160;
      }
   }
   interface fe-1/2/1.0;
   interface lo0.0;
   interface ge-0/3/0.0;
   interface ms-1/1/0.2;
}
```

As with any filter-based forwarding configuration, in order for the static route in the forwarding instance **stage** to have a reachable next hop, you must configure routing table groups so that all interface routes are copied from **inet.0** to the routing table in the forwarding instance. You configure routing tables **inet.0** and **stage.inet.0** as members of **fbf_rib_group**, so that all interface routes are imported into both tables.

```
[edit routing-options]
```



```
interface-routes {
   rib-group inet fbf_rib_group;
}
rib-groups fbf_rib_group {
   import-rib [ inet.0 stage.inet.0 ];
}
multicast {
   rpf-check-policy no_rpf;
}
```

Reverse path forwarding (RPF) checking must be disabled for the multicast group on which source NAT is applied. You can disable RPF checking for specific multicast groups by configuring a policy similar to the one in the example that follows. In this case, the **no_rpf** policy disables RPF check for multicast groups belonging to **224.0.0.0/4**.

```
[edit policy-options]
policy-statement no_rpf {
   term 1 {
      from {
         route-filter 224.0.0.0/4 orlonger;
      }
      then reject;
   }
}
```

Router 1 Configuration

The Internet Group Management Protocol (IGMP), OSPF, and PIM configuration on Router 1 is as follows. Because of IGMP static group configuration, traffic is forwarded out **fe-3/0/0.0** to the multicast receiver without receiving membership reports from host members.

```
[edit protocols]
igmp {
   interface fe-3/0/0.0 {
   }
}
ospf {
   area 0.0.0.0 {
      interface fe-3/0/0.0 {
         passive;
      }
      interface lo0.0;
      interface ge-7/2/0.0;
   }
   pim {
      rp {
         static {
            address 10.255.14.160;
         }
      }
      interface fe-3/0/0.0;
      interface lo0.0;
      interface ge-7/2/0.0;
   }
}
```

The routing option creates a static route to the NAT pool, **mcast_pool**, on the RP.

```
[edit routing-options]
static {
   route 20.20.20.0/27 next-hop 10.10.1.1;
}
```

Example: Configuring Twice NAT

In the following configuration, **term1** configures source address translation and destination address translation for traffic to a specific destination address from a source address in a range of source addresses. Both destination and source pools are configured.

```
[edit services nat]
rule twice-nat {
   match-direction input;
   term my-term1 {
      from {
         destination-address {
            41.41.41.41/32;
         }
         source-address-range {
            low 10.58.254.34 high 10.58.254.35;
         }
      }
      then {
         translated {
            source-pool src-pool;
            destination-pool dst_pool;
            translation-type {
               source static;
               destination static;
            }
         }
      }
   }
}
```

NOTE: Starting with Junos OS Release 11.2, the translation types **destination static** and **source static** are deprecated. However, these statements will continue to be supported until Junos OS Release 11.4. Therefore, you can continue to configure Twice NAT using these deprecated statements. The new statement for configuring Twice NAT will be introduced in a future release.

Example: Configuring Port Forwarding with Twice NAT

The following example configures port forwarding with **twice-napt-44** as the translation type. The example also has stateful firewall and multiple port maps configured.

```
[edit services]
user@host# show
```

```
service-set in {
    syslog {
        host local {
            services any;
        }
    }
    stateful-firewall-rules r;
    nat-rules r;
    interface-service {
        service-interface sp-10/0/0.0;
    }
}
stateful-firewall {
    rule r {
        match-direction input;
        term t {
            from {
                destination-port {
                    range low 1 high 57000;
                }
            }
            then {
                reject;
            }
        }
    }
}
nat {
    pool x {
        address 12.0.0.2/32;
    }
    rule r {
        match-direction input;
        term t {
            from {
                destination-address {
                    14.0.0.2/32;
                }
                destination-port {
                    range low 10 high 20000;
                }
            }
            then {
                port-forwarding-mappings y;
             translated {
                    destination-pool x;
                    translation-type {
                        twice-napt-44;
                    }
                }
            }
        }
    }
    port-forwarding y {
        destined-port 45;
        translated-port 23;
        destined-port 55;
        translated-port 33;
        destined-port 65;
        translated-port 43;
   }
```

```
}
adaptive-services-pics {
    traceoptions {
        file sp-trace;
        flag all;
    }
}
```

NOTE:

- Stateful firewall has precedence over port forwarding. In this example, for instance, no traffic destined to any port between 1 and 57000 will be translated.
- Up to 32 port maps can be configured.

Related Documentation

- Configuring Port Forwarding for Static Destination Address Translation on page 66

Example: NAT 44 CGN Configurations

This example describes how to implement several NAT configurations.

- Hardware and Software Requirements on page 108
- Overview on page 108
- Basic NAT44 Configuration on page 108

Hardware and Software Requirements

This example requires the following hardware:

- An MX Series 3D Universal Edge router with a Services DPC or an M Series Multiservice Edge router with a services PIC
- A domain name server (DNS)

This example uses the following software:

- Junos OS Release 11.4 or higher

Overview

This example shows a complete CGN NAT44 configuration and advanced options.

Basic NAT44 Configuration

Chassis Configuration

Step-by-Step Procedure

To configure the service PIC (FPC 5 Slot 0) with the layer 3 service package:

1. Go to the **edit chassis** hierarchy level.

   user@host# edit chassis

2. Configure the layer 3 service package.

```
[edit chassis]
user@host# set fpc 5 pic 0 adaptive-services service-package layer-3
```

Configuring the Interfaces

Step-by-Step Procedure

To configure interfaces to the private network and the public Internet:

1. Define the interface to the private network.

```
user@host# edit interfaces ge-1/3/5
[edit interfaces ge-1/3/5]
user@host# set description “Private”
user@host# edit unit 0 family inet
[edit interfaces ge-1/3/5 unit 0 family inet]
user@host# set service input service-set ss2
user@host# set service output service-set ss2
user@host# set address 9.0.0.1/24
```

2. Define the interface to the public Internet.

```
user@host# edit interfaces ge-1/3/6
[edit interfaces ge-1/3/6]
user@host# set description “Public”
user@host# set unit 0 family inet address 128.0.0.1/24
```

3. Define the service interface for NAT processing.

```
user@host# edit interfaces ge-5/0/0
[edit interfaces ge-5/0/0]
user@host# set unit 0 family inet
```

Results

```
user@host# show interfaces ge-1/3/5
description Private;
unit 0 {
    family inet {
        service {
            input {
                 service-set sset2;
            }
            output {
                service-set sset2;
            }
        }
        address 9.0.0.1/24;
    }
}
}
```

```
user@host# show interfaces ge-1/3/6
description Public:;
unit 0 {
    family inet {
        address 128.0.0.1/24;
    }
}
```

```
user@host# show interfaces ge-5/0/0
```

```
unit 0 {
family inet;
}
```

Configuring NAT with Port Translation

Step-by-Step Procedure

To configure source-only dynamic NAT with port translation:

1. Configure the NAT pool.

   ```
   user@host# edit services nat
   [edit services nat]
   user@host# set pool p1 address 129.0.0.0/24
   user@host# set pool p1 port automatic random-allocation
   ```

2. Configure the NAT rule.

   ```
   [edit services nat]

   host# edit rule r1

   host# set match-direction input

   host# set term t1 from source-address 10.0.0.0/16

   host# set term t1 from source-address 10.1.0.0/16

   host# set term t1 then translated source-pool p1 translation-type dynamic-nat44
   ```

Results

```
user@host# show services nat
pool p1 {
    address 129.0.0.0/24;
}
rule r1 {
    match-direction input;
    term t1 {
        from {
            source-address {
                10.0.0.0/16;
                10.1.0.0/16;
            }
        }
        then {
            translated {
                source-pool p1;
                translation-type {
                    dynamic-nat44;
                }
            }
        }
    }
}
```

Configuring the Service Set

Step-by-Step Procedure

To configure the service set:

1. Configure a service set.

   ```
   user@host# edit services service-set ss2
   ```

2. Specify the NAT rule to be used.

   [edit services service-set ss2]
   host# set nat-rules r1

3. Specify the interface service.

   [edit services service-set ss2]
   host# set interface-service service-interface sp-5/0/0

Results **user@host# show services service-sets sset2**

```
nat-rules r1;
interface-service {
    service-interface sp-5/0/0;
}
```

Example: NAT Between VRFs Configuration

The following example configuration enables NAT between VRFs with overlapping private addresses, using distinct public addresses for the source and destination NAT in this scenario:

- A host in **vrf-a** traverses **10.58.16.201** to reach **10.58.0.2** in **vrf-b**.
- A host in **vrf-b** traverses **10.58.16.101** to reach **10.58.0.2** in **vrf-a**.

```
[edit interfaces]
ge-0/2/0 {
  unit 0 {
    family inet {
      address 10.58.0.1/24;
      service {
        input service-set vrf-a-svc-set;
        output service-set vrf-a-svc-set;
      }
    }
  }
}
ge-0/3/0 {
  unit 0 {
    family inet {
      address 10.58.0.1/24;
      service {
        input service-set vrf-b-svc-set;
        output service-set vrf-b-svc-set;
      }
    }
  }
}
sp-1/3/0 {
  unit 0 {
    family inet;
  }
  unit 10 {
    family inet;
```

```
      service-domain inside;
    }
    unit 20 {
      family inet;
      service-domain inside;
    }
}
[edit policy-options]
policy-statement test-policy {
    term t1 {
      then reject;
    }
}
[edit routing-instances]
vrf-a {
    interface ge-0/2/0.0;
    interface sp-1/3/0.10;
    instance-type vrf;
    route-distinguisher 10.1.1.1:1;
    vrf-import test-policy;
    vrf-export test-policy;
    routing-options {
      static {
        route 0.0.0.0/0 next-table inet.0;
      }
    }
}
vrf-b {
    interface ge-0/3/0.0;
    interface sp-1/3/0.20;
    instance-type vrf;
    route-distinguisher 10.2.2.2:2;
    vrf-import test-policy;
    vrf-export test-policy;
    routing-options {
      static {
        route 0.0.0.0/0 next-table inet.0;
      }
    }
}
[edit services]
stateful-firewall {
    rule allow-all {
      match-direction input-output;
      term t1 {
        then {
          accept;
        }
      }
    }
}
nat {
    pool vrf-a-src-pool {
      address 10.58.16.100;
      port automatic;
    }
```

```
pool vrf-a-dst-pool {
    address 10.58.0.2;
}
rule vrf-a-input {
    match-direction input;
    term t1 {
        then {
            translated {
                source-pool vrf-a-src-pool;
                translation-type napt-44;
            }
        }
    }
}
rule vrf-a-output {
    match-direction output;
    term t1 {
        from {
            destination-address 10.58.16.101;
        }
        then {
            translated {
                destination-pool vrf-a-dst-pool;
                translation-type destination static;
            }
        }
    }
}
pool vrf-b-src-pool {
    address 10.58.16.200;
    port automatic;
}
pool vrf-b-dst-pool {
    address 10.58.0.2;
}
rule vrf-b-input {
    match-direction input;
    term t1 {
        then {
            translated {
                source-pool vrf-b-src-pool;
                translation-type source dynamic;
            }
        }
    }
}
rule vrf-b-output {
    match-direction output;
    term t1 {
        from {
            destination-address 10.58.16.201;
        }
        then {
            translated {
                destination-pool vrf-b-dst-pool;
                translation-type destination static;
```

```
            }
          }
        }
      }
    }
    service-set vrf-a-svc-set {
      stateful-firewall-rules allow-all;
      nat-rules vrf-a-input;
      nat-rules vrf-a-output;
      interface-service {
        service-interface sp-1/3/0.10;
      }
    }
    service-set vrf-b-svc-set {
      stateful-firewall-rules allow-all;
      nat-rules vrf-b-input;
      nat-rules vrf-b-output;
      interface-service {
        service-interface sp-1/3/0.20;
      }
    }
```

Example: Configuring Stateful NAT64 for Handling IPv4 Address Depletion

This example configures stateful NAT64 on an MX Series 3D Universal Edge router with a Services DPC. The configuration replicates the example flow found in draft-ietf-behave-v6v4-xlate-stateful-12, *Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers*, July 2010.

This example contains the following sections:

- Requirements on page 114
- Implementation on page 114
- Configuration on page 115
- Verifying NAT64 Operation on page 119

Requirements

This functionality requires the following hardware:

- An MX Series 3D Universal Edge router with a Services DPC or an M Series Multiservice Edge router with a services PIC
- A name server with DNS64

Implementation

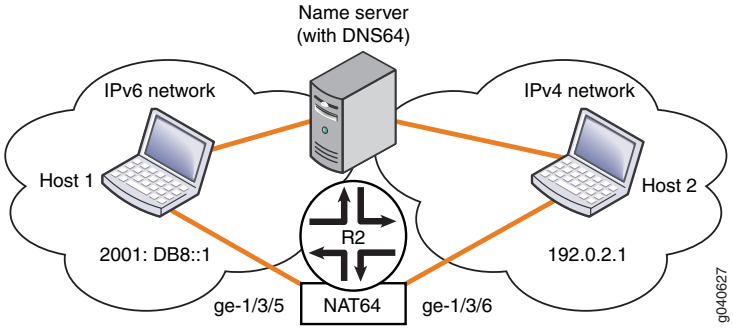
In Junos OS Release 10.2, Juniper Networks implemented stateful NAT64 in its Services PIC and Services Dense Port Concentrator (DPC). The system steers IPv6 packets coming from IPv6-only hosts to a Services DPC where the packets are translated to IPv4 according to the configuration. In the reverse path, the system sends IPv4 packets to the Services DPC where additional system processes reverse the translation and send the corresponding IPv6 packet back to the client.

Configuration Overview and Topology

Figure 9 on page 115 shows an MX Series router, R2, implementing NAT64 with two Gigabit Ethernet interfaces and a Services DPC. The interface connected to the IPv4 network is **ge-1/3/6**, and the interface connected to the IPv6 network is **ge-1/3/5**.

Also shown is a local name server with DNS64 functionality, which the system uses as part of the translation process. The local name server is configured with the **/96** prefix assigned to the local NAT64 router.

Figure 9: NAT64 Topology



Configuration

To configure stateful NAT64 involves the following tasks:

- Configuring the PIC and the Interfaces on page 115
- Configuring the NAT64 Pool on page 117
- Configuring the Service Set on page 118

Configuring the PIC and the Interfaces

Step-by-Step Procedure

To configure the PIC and interfaces on Router R2:

1. Edit the **chassis** configuration to enable a Layer 3 service package. The service package with its associated service package (**sp-**) interface is used to manipulate traffic before it is delivered to its destination. For details about configuring packages, see the *Junos OS Services Interfaces Configuration Guide*.
2. Configure the service package at the **[edit chassis fpc pic adaptive-services]** hierarchy level. This example assumes that the PIC is in FPC 5, slot 0.

```
[edit chassis]
fpc 5 {
  pic 0 {
    adaptive-services {
      service-package layer-3;
    }
  }
}
```

3. Configure the **ge-1/3/5** interface connected to the IPv6 network:

   a. Include the **family inet** (IPv4) and **family inet6** (IPv6) statements at the **[edit interfaces *interface-name* unit *unit-number*]** hierarchy level.

   b. Include the IPv6 address at the **[edit interfaces unit *unit-number* family inet6 address]** hierarchy level.

   c. Configure a service set at the **[edit interfaces *interface-name* unit *unit-number* family service input service-set]** and the **[edit interfaces *interface-name* unit *unit-number* family service output service-set]** hierarchy levels.

```
[edit interfaces]
ge-1/3/5 {
  description "IPv6-only domain";
  unit 0 {
    family inet;
    family inet6 {
      service {
        input {
          service-set set_0;
        }
        output {
          service-set set_0;
        }
      }
      address 2001:DB8::1/64;
    }
  }
}
```

4. Configure the **ge-1/3/6** interface connected to the IPv4 network:

   a. Include the **family inet** statement at the **[edit interfaces unit *unit-number*]** hierarchy level.

   b. Include the IPv4 address at the **[edit interfaces unit *unit-number* family inet]** hierarchy level.

```
[edit interfaces]
ge-1/3/6 {
  description "Internet-IPv4 domain";
  unit 0 {
    family inet {
      address 192.0.1.1/16;
    }
  }
}
```

5. Configure the services interface, in this example, **sp-5/0/0**. This example configures a system log for any services on the local host.

   The service package associated with this interface was configured in Step 2. Specify both the IPv4 and IPv6 address families at the **[edit interfaces *interface-name* unit *unit-number*]** hierarchy level. The service set you configure in "Configuring the Service Set" on page 118 is associated with this interface.


```
[edit interfaces]
sp-5/0/0 {
   services-options {
      syslog {
         host local {
            services any;
            log-prefix XXXXXXXX;
         }
      }
   }
   unit 0 {
      family inet;
      family inet6;
   }
}
```

Configuring the NAT64 Pool

Step-by-Step Procedure

Use this procedure to configure the NAT64 router, Router R2, with the **/96** prefix to represent IPv4 addresses in the IPv6 address space. IPv6 packets addressed to a destination address containing the **/96** prefix are then routed to the IPv6 interface of the NAT router. You also configure one or more IPv4 transport addresses for the NAT pool.

This example shows how to configure the network address translation for the IPv4 address **203.0.113.1/32**. It also shows how to configure the IPv6 prefix **64:FF9B::/96**. To configure the NAT64 pool:

1. Configure an IPv4 transport address for the pool at the **[edit services nat pool *pool-name*]** hierarchy level.

   ```
   [edit services nat]
   pool src-pool-nat64 {
      address 203.0.113.0/24;
      port automatic;
   }
   ```

2. Configure a NAT rule to translate the packets from the IPv6 network. NAT rules specify the traffic to be matched and the action to be taken when traffic matches the rule.

   In this example, only one rule is required to accomplish the address translation. The rule selects all traffic coming from the source address on the IPv6 network, **2001:DB8::1/128**. The transport address configured in Step 1 is then specified for the translation using the **/96** prefix.

   Configure the rule at the **[edit services nat rule *rule-name*]** hierarchy level as follows:

   ```
   [edit services nat rule]
   rule nat64 {
      match-direction input;
      term t1 {
         from {
            source-address {
               2001:DB8::0/96;
            }
   ```

```
        destination-address {
          64:FF9B::/96;
        }
      }
      then {
        translated {
          source-pool src-pool-nat64;
          destination-prefix 64:FF9B::/96;
          translation-type {
            stateful-nat64;
          }
        }
      }
    }
  }
```

Configuring the Service Set

Step-by-Step Procedure

To configure the service set for the NAT service on Router R2, you must associate the previously configured rule (**nat64**) and service interface (**sp-5/0/0**) with the service set. You also include a system log configuration.

To configure these settings at the **[edit services service-set *service-set-name*]** hierarchy level:

1. Configure the system log.

```
[edit services service-set set_0]
syslog {
  host local {
    services any;
    log-prefix XXXSVC-SETYYY;
  }
}
```

2. Associate the NAT rule and the service interface with the service set at the **[edit services service-set *service-set-name*]** hierarchy level.

```
[edit services ]
service-set {
  nat-rules nat64;
  interface-service {
    service-interface sp-5/0/0;
  }
}
```

3. On Router R2, commit the configuration.

```
user@R2> commit check
configuration check succeeds
user@R2> commit
```

Verifying NAT64 Operation

You can use the following features to verify your NAT64 configuration:

- CLI commands on the router
- Logging

You can also use a test tool that can generate IPv6 flows directed to the MX Series router, using the well-known prefix (**64:FF9B::/96**) as the destination.

NAT64-related commands leverage the existing commands for NAPT44.

Among others, you can use the following CLI commands to verify your NAT64 configuration:

- **show services stateful-firewall flows**
- **show services stateful-firewall conversations**
- **show services nat pool detail**
- **show services stateful-firewall statistics extensive**

In this example:

- In the input direction, the IPv4 destination address is fetched from the IPv6 destination address whose prefix matches the destination-prefix configured from the specified prefix length.
- In the reverse or output direction, the IPv4 address is suffixed to the destination-prefix at the prefix length specified.

To confirm the NAT64 configuration, perform these tasks:

- Display NAT64 Flows on page 119
- Display NAT64 Conversations on page 120
- Display Global NAT Pool-Related Statistics on page 122
- Check System Logs on page 122
- Verify That NAT64 Conversations Take Place on page 123

Display NAT64 Flows

Purpose Display and verify that the NAT64 flows are created and contain correct network address translation.

Action To display the NAT64 flows on Router R2, use the **show services stateful-firewall flows** command.

```
user@R2> show services stateful-firewall flows

Interface: sp-5/0/0, Service set: set_0
Flow                                                  State      Dir     Frm count
TCP        2001:db8::4:1160  ->64:ff9b::c000:201:80     Forward    I              5
```

```
    NAT source      2001:db8::4:1160     ->      203.0.113.1:
    NAT dest    64:ff9b::c000:201:80     ->        192.0.2.1:80
TCP         2001:db8::2:1166  ->64:ff9b::c000:201:80     Forward    I              5
    NAT source      2001:db8::2:1166     ->      203.0.113.1:1420
    NAT dest    64:ff9b::c000:201:80     ->        192.0.2.1:80
TCP           192.0.2.1:80     ->    203.0.113.1:1413     Forward    O              4
    NAT source        192.0.2.1:80       -> 64:ff9b::c000:201:21286
    NAT dest        203.0.113.1:1413     ->      2001:db8::4:1167
TCP         2001:db8::3:1123  ->64:ff9b::c000:201:80     Forward    I              5
    NAT source      2001:db8::3:1123     ->      203.0.113.1:1385
    NAT dest    64:ff9b::c000:201:80     ->        192.0.2.1:80
TCP           192.0.2.1:80     ->    203.0.113.1:1376     Forward    0              4
    NAT source        192.0.2.1:80       -> 64:ff9b::c000:201:21367
    NAT dest        203.0.113.1:1376     ->      2001:db8::3:1120
TCP         2001:db8::3:1136  ->64:ff9b::c000:201:80     Forward    I              5
    NAT source      2001:db8::3:1136     ->      203.0.113.1:1424
    NAT dest    64:ff9b::c000:201:80     ->        192.0.2.1:80
TCP         2001:db8::4:1146  ->64:ff9b::c000:201:80     Forward    I              5
    NAT source      2001:db8::4:1146     ->      203.0.113.1:1350
    NAT dest    64:ff9b::c000:201:80     ->        192.0.2.1:80
TCP         2001:db8::3:1110  ->64:ff9b::c000:201:80     Forward    I              5
    NAT source      2001:db8::3:1110     ->      203.0.113.1:1346
    NAT dest    64:ff9b::c000:201:80     ->        192.0.2.1:80
TCP           192.0.2.1:80     ->    203.0.113.1:1428     Forward    0              4
    NAT source        192.0.2.1:80       -> 64:ff9b::c000:201:21367
    NAT dest        203.0.113.1:1428     ->      2001:db8::4:1172
TCP           192.0.2.1:80     ->    203.0.113.1:1393     Forward    0              4
    NAT source        192.0.2.1:80       -> 64:ff9b::c000:201:80
    NAT dest        203.0.113.1:1393     ->      2001:db8::2:1157
TCP           192.0.2.1:80     ->    203.0.113.1:1346     Forward    0              4
    NAT source        192.0.2.1:80       -> 64:ff9b::c000:201:21367
    NAT dest        203.0.113.1:1346     ->      2001:db8::3:1110
TCP         2001:db8::2:1148  ->64:ff9b::c000:201:80     Forward    I              5
    NAT source      2001:db8::2:1148     ->      203.0.113.1:1366
    NAT dest    64:ff9b::c000:201:80     ->        192.0.2.1:80
TCP           192.0.2.1:80     ->    203.0.113.1:1363     Forward    0              4
```

Meaning In the sample output, the NAT source and NAT destination addresses of the Input (I) and Output (O) directions are displayed. The NAT64 flows listed in this output are in no specific order.

Display NAT64 Conversations

Purpose Display and verify that the NAT64 conversations (collections of related flows) are correct.

Action To display NAT64 conversations on Router R2, use the **show services stateful-firewall conversations** command. In contrast to the **flows** command that reports all flows in no specific order, the output of the **conversations** command groups the flows that belong to a conversation for easy troubleshooting of communication between a specific pair of hosts.

```
user@R2> show services stateful-firewall conversations

Interface: sp-5/0/0, Service set: set_0

Conversation: ALG protocol: tcp
  Number of initiators: 1, Number of responders: 1
Flow                                          State    Dir       Frm count
```

```
TCP        2001:db8::3:1188  ->64:ff9b::c000:201:80  Forward  I               5
    NAT source     2001:db8::3:1188    ->     203.0.113.1:1580
    NAT dest   64:ff9b::c000:201:80    ->       192.0.2.1:80
TCP          192.0.2.1:80    ->    203.0.113.1:1580  Forward  O               4
    NAT source       192.0.2.1:80      -> 64:ff9b::c000:201:21303
    NAT dest       203.0.113.1:1580    ->     2001:db8::3:1188

Conversation: ALG protocol: tcp
  Number of initiators: 1, Number of responders: 1
Flow                                                 State    Dir       Frm count
TCP        2001:db8::4:1213  ->64:ff9b::c000:201:80  Forward  I               5
    NAT source     2001:db8::4:1213    ->     203.0.113.1:1551
    NAT dest   64:ff9b::c000:201:80    ->       192.0.2.1:80
TCP          192.0.2.1:80    ->    203.0.113.1:1551  Forward  O               4
    NAT source       192.0.2.1:80      -> 64:ff9b::c000:201:21367
    NAT dest       203.0.113.1:1551    ->     2001:db8::4:1213

Conversation: ALG protocol: tcp
  Number of initiators: 1, Number of responders: 1
Flow                                                 State    Dir       Frm count
TCP        2001:db8::3:1169  ->64:ff9b::c000:201:80  Forward  I               5
    NAT source     2001:db8::3:1169    ->     203.0.113.1:1523
    NAT dest   64:ff9b::c000:201:80    ->       192.0.2.1:80
TCP          192.0.2.1:80    ->    203.0.113.1:1523  Forward  O               4
    NAT source       192.0.2.1:80      -> 64:ff9b::c000:201:80
    NAT dest       203.0.113.1:1523    ->     2001:db8::3:1169

Conversation: ALG protocol: tcp
  Number of initiators: 1, Number of responders: 1
Flow                                                 State    Dir       Frm count
TCP        2001:db8::2:1233  ->64:ff9b::c000:201:80  Forward  I               5
    NAT source     2001:db8::2:1233    ->     203.0.113.1:1621
    NAT dest   64:ff9b::c000:201:80    ->       192.0.2.1:80
TCP          192.0.2.1:80    ->    203.0.113.1:1621  Forward  O               4
    NAT source       192.0.2.1:80      -> 64:ff9b::c000:201:21367
    NAT dest       203.0.113.1:1621    ->     2001:db8::2:1233

Conversation: ALG protocol: tcp
  Number of initiators: 1, Number of responders: 1
Flow                                                 State    Dir       Frm count
TCP        2001:db8::2:1218  ->64:ff9b::c000:201:80  Forward  I               5
    NAT source     2001:db8::2:1218    ->     203.0.113.1:1575
    NAT dest   64:ff9b::c000:201:80    ->       192.0.2.1:80
TCP          192.0.2.1:80    ->    203.0.113.1:1575  Forward  O               4
    NAT source       192.0.2.1:80      -> 64:ff9b::c000:201:21367
    NAT dest       203.0.113.1:1575    ->     2001:db8::2:1218

Conversation: ALG protocol: tcp
  Number of initiators: 1, Number of responders: 1
Flow                                                 State    Dir       Frm count
TCP        2001:db8::4:1220  ->64:ff9b::c000:201:80  Forward  I               5
    NAT source     2001:db8::4:1220    ->     203.0.113.1:1572
    NAT dest   64:ff9b::c000:201:80    ->       192.0.2.1:80
TCP          192.0.2.1:80    ->    203.0.113.1:1572  Forward  O               4
    NAT source       192.0.2.1:80      -> 64:ff9b::c000:201:21367
    NAT dest       203.0.113.1:1572    ->     2001:db8::4:1220

Conversation: ALG protocol: tcp
  Number of initiators: 1, Number of responders: 1
Flow                                                 State    Dir       Frm count
TCP        2001:db8::2:1211  ->64:ff9b::c000:201:80  Forward  I               5
```

```
    NAT source     2001:db8::2:1211    ->     203.0.113.1:1554
    NAT dest   64:ff9b::c000:201:80    ->       192.0.2.1:80
TCP          192.0.2.1:80    ->    203.0.113.1:1554  Forward  0              4
    NAT source       192.0.2.1:80      -> 64:ff9b::c000:201:21286
    NAT dest       203.0.113.1:1554    ->     2001:db8::2:1211
```

Meaning The sample output displays the NAT64 conversations between specific pairs of hosts.

Display Global NAT Pool-Related Statistics

Purpose Display and verify global NAT statistics related to pool usage.

Action To display global NAT pool-related statistics on Router R2, use the **show services nat pool detail** command. You normally use this command in conjunction with the **show services stateful-firewall flows** command used in "Display NAT64 Flows" on page 119, which displays the source and output of the translation.

```
user@R2> show services nat pool detail

Interface: sp-5/0/0, Service set: set_0
  NAT pool: src-pool-nat64, Translation type: dynamic
    Address range: 203.0.113.1-203.0.113.254
    Port range: 512-65535, Ports in use: 102, Out of port errors: 0, Max ports used: 192
  NAT pool: _jpool_nat64_t1_, Translation type: static
    Address range: 0.100.255.155-0.100.255.154
```

Meaning The sample output displays relevant statistics and information about the NAT64 pools.

Check System Logs

Purpose Check the system logs because the system creates detailed logs as sessions are created and deleted.

Action When a session is created based on the example setup, two logs are provided. The first log indicates the rule and term that the packet matched. The second log indicates the flow creation.

```
user@R2> show log messages
Oct 21 22:14:14  H1 (FPC Slot 5, PIC Slot 0) XXXSVC-SETYYY{set_0}[FWNAT]:
ASP_SFW_CREATE_ACCEPT_FLOW: proto 6 (TCP) application: any,
ge-1/3/5.0:2001:db8:0:0:0:0:0:1:1025 -> 64:ff9b:0:0:0:0:c000:201:80, creating
forward or watch flow ; source address and port translate to 203.0.113.1:1593 ;
destination address translates to 192.0.2.1
```

When the sessions end, the system creates a log indicating the NAT pool address and port release in addition to the delete flow log, as follows:

```
Oct 21 22:14:17  H1 (FPC Slot 5, PIC Slot 0)
XXXSVC-SETYYY{set_0}[FWNAT]:ASP_NAT_POOL_RELEASE: natpool release
203.0.113.1:1593[1]
Oct 21 22:14:17  H1 (FPC Slot 5, PIC Slot 0) XXXSVC-SETYYY{set_0}[FWNAT]:
ASP_SFW_DELETE_FLOW: proto 6 (TCP) application: any,
(null)(null)2001:db8:0:0:0:0:0:1:1025 -> 64:ff9b:0:0:0:0:c000:201:80, deleting
```

```
forward or watch flow ; source address and port translate to 203.0.113.1:1593 ;
destination address translates to 192.0.2.1
```

Meaning The sample output displays the log messages that can be seen when a session is created and when a session ends.

Verify That NAT64 Conversations Take Place

Purpose Verify that the NAT64 conversations are taking place. Current support for the application-layer gateway (ALG) is limited to ICMP and traceroute.

Action To verify that the NAT64 conversations are occuring on Router R2, use the **show services stateful-firewall conversations** command. The following is sample output for an ICMP echo test (**ping**).

```
user@R2> show services stateful-firewall conversations

Interface: sp-5/0/0, Service set: set_0

Conversation: ALG protocol: icmpv6
  Number of initiators: 1, Number of responders: 1
Flow                                        State    Dir       Frm count
ICMPV6     2001:db8::2       ->64:ff9b::c000:201     Watch    I              21
    NAT source     2001:db8::2         ->     203.0.113.1
    NAT dest   64:ff9b::c000:201       ->       192.0.2.1
ICMP         192.0.2.1       ->    203.0.113.1       Watch    0              21
    NAT source       192.0.2.1         -> 64:ff9b::c000:201
    NAT dest       203.0.113.1         ->     2001:db8::2
```

Meaning The sample output displays the results of the ICMP echo test.

Related Documentation

- Stateful NAT64 Overview
- Example: Configuring Dual-Stack Lite for IPv6 Access

CHAPTER 6

NAT Configuration Statements

address

Syntax address *ip-prefix</prefix-length>*;

Hierarchy Level [edit services nat pool *nat-pool-name*]

Release Information Statement introduced before Junos OS Release 7.4.
prefix option enhanced to support IPv6 addresses in Junos OS Release 8.5.

Description Specify the NAT pool prefix value.

Options ***prefix***—Specify an IPv4 or IPv6 prefix value.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Configuring Addresses and Ports for Use in NAT Rules on page 21

address-allocation

Syntax address-allocation round-robin;

Hierarchy Level [edit services nat pool *pool-name*]

Release Information Statement introduced in Junos OS Release 11.2.

Description When you use round-robin allocation, one port is allocated from each address in a range before repeating the process for each address in the next range. After ports have been allocated for all addresses in the last range, the allocation process wraps around and allocates the next unused port for addresses in the first range.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Configuring Addresses and Ports for Use in NAT Rules on page 21

address-range

Syntax address-range low *minimum-value* high *maximum-value*;

Hierarchy Level [edit services nat pool *nat-pool-name*]

Release Information Statement introduced before Junos OS Release 7.4.
minimum-value and ***maximum-value*** options enhanced to support IPv6 addresses in Junos OS Release 8.5.

Description Specify the NAT pool address range.

Options ***minimum-value***—Lower boundary for the IPv4 or IPv6 address range.

maximum-value—Upper boundary for the IPv4 or IPv6 address range.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Configuring Addresses and Ports for Use in NAT Rules on page 21

allow-overlapping-nat-pools

Syntax allow-overlapping-nat-pools;

Hierarchy Level [edit services nat]

Release Information Statement introduced with Junos OS Release 12.1.

Description Specify that NAT source pools can be shared between multiple service sets.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Configuring NAT Service Sets on page 36

application-sets

Syntax applications-sets *set-name*;

Hierarchy Level [edit services nat rule *rule-name* term *term-name* from (Services NAT)]

Release Information Statement introduced before Junos OS Release 7.4.

Description Define one or more target application sets.

Options ***set-name***—Name of the target application set.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Configuring Match Conditions in NAT Rules on page 32

applications

Syntax applications [*application-names*];

Hierarchy Level [edit services nat rule *rule-name* term *term-name* from (Services NAT)]

Release Information Statement introduced before Junos OS Release 7.4.

Description Define one or more application protocols to which the NAT services apply.

Options ***application-name***—Name of the target application.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Configuring Match Conditions in NAT Rules on page 32

cgn-pic

Syntax cgn-pic;

Hierarchy Level [edit interfaces *interface-name* services-options]

Release Information Statement introduced in Junos OS Release 11.2.

Description Restrict usage of the service PIC to carrier-grade NAT (CGN) or associated services (intrusion detection, stateful firewall, and softwire). All memory is available for CGN or related services and can be used for CGN scaling.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Configuring NAT Service Sets on page 36

destination-address

Syntax destination-address (*address* | any-unicast) <except>;

Hierarchy Level [edit services nat rule *rule-name* term *term-name* from (Services NAT)]

Release Information Statement introduced before Junos OS Release 7.4.
any-unicast and **except** options introduced in Junos OS Release 7.6.
address option enhanced to support IPv6 and addresses in Junos OS Release 8.5.

Description Specify the destination address for rule matching.

Options ***address***—Destination IPv4 or IPv6 address or prefix value.

any-unicast—Any unicast packet.

except—(Optional) Prevent the specified address, prefix, or unicast packets from being translated.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Configuring Match Conditions in NAT Rules on page 32

destination-address-range

Syntax destination-address-range low *minimum-value* high *maximum-value* <except>;

Hierarchy Level [edit services nat rule *rule-name* term *term-name* from (Services NAT)]

Release Information Statement introduced in Junos OS Release 7.6.
minimum-value and ***maximum-value*** options enhanced to support IPv6 addresses in Junos OS Release 8.5.

Description Specify the destination address range for rule matching.

Options ***minimum-value***—Lower boundary for the IPv4 or IPv6 address range.

maximum-value—Upper boundary for the IPv4 or IPv6 address range.

except—(Optional) Prevent the specified address range from being translated.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Configuring Match Conditions in NAT Rules on page 32

destination-pool

Syntax destination-pool *nat-pool-name*;

Hierarchy Level [edit services nat rule *rule-name* term *term-name* then translated]

Release Information Statement introduced before Junos OS Release 7.4.

Description Specify the destination address pool for translated traffic.

Options ***nat-pool-name***—Destination pool name.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Configuring Actions in NAT Rules on page 33

destination-port range

Syntax destination-port range *high* | *low*;

Hierarchy Level [edit services nat rule *rule-name* term *term-name* from (Services NAT)]

Release Information Statement introduced in Junos OS Release 11.4.

Description Specify the destination port range for rule matching.

Options ***high***—Upper limit of port range for matching.

low—Lower limit of port range for matching.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Configuring Port Forwarding for Static Destination Address Translation on page 66

destination-prefix

Syntax destination-prefix *destination-prefix*;

Hierarchy Level [edit services nat rule *rule-name* term *term-name* then translated]

Release Information Statement introduced in Junos OS Release 7.6.
destination-prefix option enhanced to support IPv6 addresses in Junos OS Release 8.5.

Description Specify the destination prefix for translated traffic.

Options ***destination-prefix***—IPv4 or IPv6 destination prefix value.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Configuring Actions in NAT Rules on page 33

destination-prefix-list

Syntax destination-prefix-list *list-name* <except>;

Hierarchy Level [edit services nat rule *rule-name* term *term-name* from (Services NAT)]

Release Information Statement introduced in Junos OS Release 8.2.

Description Specify the destination prefix list for rule matching. You configure the prefix list by including the **prefix-list** statement at the **[edit policy-options]** hierarchy level.

Options ***list-name***—Destination prefix list.

except—(Optional) Exclude the specified prefix list from rule matching.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Configuring Match Conditions in NAT Rules on page 32
- *Junos OS Policy Framework Configuration Guide*

destined-port

Syntax destined-port *port id*;

Hierarchy Level [edit services nat port-forwarding *map-name*]

Release Information Statement introduced in Junos OS Release 11.4.

Description Specify the port from where traffic has to be forwarded.

Options ***port id***—The destination port number from where traffic will be forwarded.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- port-forwarding on page 141
- translated-port on page 149

deterministic-port-block-allocation

Syntax
```
deterministic-port-block-allocation {
  block-size block-size;
}
```

Hierarchy Level [edit services nat pool *pool-name* port]

Release Information Statement introduced in Junos OS Release 12.1.

Description Configure algorithm-based allocation of blocks of destination ports. By specifying this method, you ensure that an incoming (source) IP address and port always map to the same destination IP address and port block, thus eliminating the need for logging address translations.

Options ***block-size***—Maximum number of blocks that can be allocated to a user. When 0 is specified, block size is calculated by dividing the number of ports available in the source-pool by the number of subscribers in the **from** statement in the applicable NAT rule.

Default: 256

Range: 0 to 512

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Configuring Addresses and Ports for Use in NAT Rules on page 21

dns-alg-pool

Syntax dns-alg-pool *dns-alg-pool*;

Hierarchy Level [edit services nat rule *rule-name* term *term-name* then translated]

Release Information Statement introduced in Junos OS Release 10.4.

Description Specify the Network Address Translation (NAT) pool for destination translation.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

dns-alg-prefix

Syntax dns-alg-prefix *dns-alg-prefix*;

Hierarchy Level [edit services nat rule *rule-name* term *term-name* then translated]

Release Information Statement introduced in Junos OS Release 10.4.

Description Set the Domain Name System (DNS) application-level gateway (ALG) 96-bit prefix for mapping IPv4 addresses to IPv6 addresses.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

from

Syntax

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

Hierarchy Level [edit services nat rule *rule-name* term *term-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Specify input conditions for the NAT term.

Options For information on match conditions, see the description of firewall filter match conditions in the *Junos OS Policy Framework Configuration Guide*.

The remaining statements are explained separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring NAT Rules on page 30

hint

Syntax hint [*hint-strings*];

Hierarchy Level [edit services nat pool *nat-pool-name* pgcp]

Release Information Statement introduced in Junos OS Release 9.0.

Description Configure a hint that enables the border gateway function (BGF) to choose a NAT pool by direction rather than by virtual interface. The BGF matches the configured hint with a termination hint located in the Direction field of a nonstandard termination ID.

Default When no hint is configured, the BGF can choose any NAT pool associated with the virtual interface.

Options ***hint-string***—Alphanumeric string of up to three characters that the BGF uses to match with a termination hint located in the Direction field of a nonstandard termination ID. You can also include underscores (_) and hyphens (-) within the string. To specify a list of hints, use the format: **[hint *xx* hint *yy*]**.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Session Border Control Solutions Guide Using BGF and IMSG*

ipv6-multicast-interfaces

Syntax

```
ipv6-multicast-interfaces (all | interface-name) {
    disable;
}
```

Hierarchy Level [edit services nat],
[edit services softwire]

Release Information Statement introduced in Junos OS Release 9.1.

Description Enable multicast filters on Ethernet interfaces when IPv6 NAT is used for neighbor discovery.

Options

all—Enable filters on all interfaces.

disable—Disable filters on the specified interfaces.

interface-name—Enable filters on a specific interface only.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring IPv6 Multicast Filters on page 21
- Configuring IPv6 Multicast Interfaces

match-direction

Syntax

```
match-direction (input | output);
```

Hierarchy Level [edit services nat rule *rule-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Specify the direction in which the rule match is applied.

Options

input—Apply the rule match on input.

output—Apply the rule match on output.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring NAT Rules on page 30

nat-type

Syntax nat-type (full-cone | symmetric);

Hierarchy Level [edit services nat rule *rule-name* term *term-name*]

Release Information Statement introduced in JUNOS Release 8.5.

Description Specify whether the term supports full-cone or traditional (symmetric) NAT.

Default symmetric

Options **full-cone**—Support full-cone NAT processing, in which all requests from the same internal IP address and port are mapped to the same external IP address and port.

symmetric—Support traditional NAT address matching only.

Usage Guidelines See "Configuring NAT Type for Terms in NAT Rules" on page 30

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

no-translation

Syntax no-translation;

Hierarchy Level [edit services nat rule *rule-name* term *term-name* then]

Release Information Statement introduced in Junos OS Release 7.6.

Description Specify that traffic is not to be translated.

Options none

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Actions in NAT Rules on page 33

overload-pool

Syntax overload-pool *overload-pool-name*;

Hierarchy Level [edit services nat rule *rule-name* term *term-name* then translated]

Release Information Statement introduced in Junos OS Release 7.6.

Description Specify an address pool that can be used if the source pool becomes exhausted.

Options ***overload-pool-name***—Name of the overload pool.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Configuring Actions in NAT Rules on page 33

overload-prefix

Syntax overload-prefix *overload-prefix*;

Hierarchy Level [edit services nat rule *rule-name* term *term-name* then translated]

Release Information Statement introduced in Junos OS Release 7.6.

Description Specify the prefix that can be used if the source pool becomes exhausted.

Options ***overload-prefix***—Prefix value.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Configuring Actions in NAT Rules on page 33

pgcp

Syntax

```
pgcp {
  hint [ hint-strings ];
  ports-per-session ports;
  remotely-controlled;
  transport [ transport-protocols ];
}
```

Hierarchy Level [edit services nat pool *nat-pool-name*]

Release Information Statement introduced in Junos OS Release 8.4.
remotely-controlled and **ports-per-session** statements added in Junos OS Release 8.5.
hint statement added in Junos OS Release 9.0.

Description Specify that the NAT pool is used exclusively by the BGF.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Session Border Control Solutions Guide Using BGF and IMSG*

pool

Syntax

```
pool nat-pool-name {
  address ip-prefix</prefix-length>;
  address-allocation round-robin;
  address-range low minimum-value high maximum-value;
  mapping-timeout seconds;
  pgcp {
    hint [ hint-strings ];
    ports-per-session ports;
    remotely-controlled:
    transport [ transport-protocols ];
  }
  port (automatic | range low minimum-value high maximum-value) {
    preserve-parity;
    preserve-range;
    secured-port-block-allocation {
      active-block-timeout timeout-seconds;
      block-size block-size;
      max-blocks-per-user max-blocks;
    }
  }
}
```

Hierarchy Level [edit services nat]

Release Information Statement introduced before Junos OS Release 7.4.
pgcp statement added in Junos OS Release 8.4.
remotely-controlled and **ports-per-session** statements added in Junos OS Release 8.5.
hint statement added in Junos OS Release 9.0.
address-allocation statement added in Junos OS Release 11.2.

Description Specify the NAT name and properties.

Options *nat-pool-name*—Identifier for the NAT address pool.

The remaining statements are explained separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Addresses and Ports for Use in NAT Rules on page 21

port

Syntax

```
port (automatic | range low minimum-value high maximum-value ,random-allocation) {
  preserve-parity;
  preserve-range;
  deterministic-port-block-allocation <block-size block-size>;
  secured-port-block-allocation {
    active-block-timeout timeout-seconds;
    block-size block-size;
    max-blocks-per-user max-blocks;
  }
}
```

Hierarchy Level [edit services nat pool *nat-pool-name*]

Release Information

port statement introduced before Junos OS Release 7.4.
random-allocation statement introduced in Junos OS Release 9.3.
secured-port-block-allocation statement introduced in Junos OS Release 11.2.
deterministic-port-block-allocation statement introduced in Junos OS Release 12.1.

Description Specify the NAT pool port or range. You can configure an automatically assigned port or specify a range with minimum and maximum values.

Options

automatic—Router-assigned port.

minimum-value—Lower boundary for the port range.

maximum-value—Upper boundary for the port range.

preserve-parity—Allocate ports with same parity as the original port.

preserve-range—Preserve privileged port range after translation.

random-allocation—Allocate ports within a specified range randomly.

Other options are described separately.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Addresses and Ports for Use in NAT Rules on page 21

port-forwarding

Syntax

```
port-forwarding map-name {
  destined-port;
  translated-port;
}
```

Hierarchy Level [edit services nat]

Release Information Statement introduced in Junos OS Release 11.4.

Description Specify the mapping for port forwarding.

Options *map-name*—Identifier for the port forwarding map.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Port Forwarding for Static Destination Address Translation on page 66
- Configuring Port Forwarding without Destination Address Translation on page 69

port-forwarding-mappings

Syntax port-forwarding-mappings *map-name*;

Hierarchy Level [edit services nat rule *rule-name* term *term-name* then]

Release Information Statement introduced in Junos OS Release 11.4.

Description Specify the name for mapping port forwarding in a Network Address Translation configuration.

Options *map-name*—Identifier for the port forwarding mapping.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Port Forwarding for Static Destination Address Translation on page 66
- Configuring Port Forwarding without Destination Address Translation on page 69

ports-per-session

Syntax ports-per-session *ports*;

Hierarchy Level [edit services nat pool *nat-pool-name* pgcp]

Release Information Statement introduced in Junos OS Release 8.4.

Description Configure the number of ports required to support Real-Time Transport Protocol (RTP), Real-Time Control Protocol (RTCP), Real-Time Streaming Protocol (RTSP), and forward error correction (FEC) for voice and video flows on the Multiservices PIC.

Options ***number-of-ports***—Number of ports to enable: 2 or 4 for combined voice and video services.
Default: 2

Required Privilege Level interface—To view this statement in the configuration.
interface–control—To add this statement to the configuration.

Related Documentation
- *Session Border Control Solutions Guide Using BGF and IMSG*

remotely-controlled

Syntax remotely-controlled;

Hierarchy Level [edit services nat pool *nat-pool-name* pgcp]

Release Information Statement introduced in Junos OS Release 8.5.

Description Configure the addresses and ports in a NAT pool to be remotely controlled by the gateway controller.

Required Privilege Level interface—To view this statement in the configuration.
interface–control—To add this statement to the configuration.

Related Documentation
- *Session Border Control Solutions Guide Using BGF and IMSG*

rule

Syntax

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

Hierarchy Level

[edit services nat],
[edit services nat rule-set *rule-set-name*]

Release Information

Statement introduced before Junos OS Release 7.4.

Description

Specify the rule the router uses when applying this service.

Options

rule-name—Identifier for the collection of terms that make up this rule.

The remaining statements are explained separately.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring NAT Rules on page 30

rule-set

Syntax

```
rule-set rule-set-name {
    [ rule rule-names ];
}
```

Hierarchy Level [edit services nat]

Release Information Statement introduced before Junos OS Release 7.4.

Description Specify the rule set the router uses when applying this service.

Options ***rule-set-name***—Identifier for the collection of rules that constitute this rule set.

Required Privilege Level
interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring NAT Rule Sets on page 35

services

Syntax services nat { .. }

Hierarchy Level [edit]

Release Information Statement introduced before Junos OS Release 7.4.

Description Define the service rules to be applied to traffic.

Options **nat**—Identifies the NAT set of rules statements.

Required Privilege Level
interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

secured-port-block-allocation

Syntax

```
secured-port-block-allocation {
  active-block-timeout timeout-seconds;
  block-size block-size;
  max-blocks-per-address max-blocks;
}
```

Hierarchy Level [edit services nat pool *pool-name* port]

Release Information Statement introduced in Junos OS Release 11.2.

Description When you use block allocation, one or more blocks of ports in a NAT pool address range are available for assignment to a subscriber.

Options

block-size—Number of ports included in a block.

Default: 128

Range: 1 to 60,000

max-blocks—Maximum number of blocks that can be allocated to a user address.

Default: 8

Range: 1 to 512

timeout-seconds—Interval, in seconds, during which a block is active. After timeout, a new block is allocated, even if ports are available in the active block.

Default: 0—The default timeout of the active block is 0 (infinite). In this case, the active block transitions to inactive only when it runs out of ports and a new block is allocated. Any inactive block without any ports in use will be freed to the NAT pool.

Range: Any value greater than or equal to 120.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Addresses and Ports for Use in NAT Rules on page 21

source-address

Syntax source-address (*address* | any-unicast) <except>;

Hierarchy Level [edit services nat rule *rule-name* term *term-name* from]

Release Information Statement introduced before Junos OS Release 7.4.
any-unicast and **except** options introduced in Junos OS Release 7.6.
address option enhanced to support IPv6 addresses in Junos OS Release 8.5.

Description Specify the source address for rule matching.

Options ***address***—Source IPv4 or IPv6 address or prefix value.

any-unicast—Any unicast packet.

except—(Optional) Prevent the specified address or unicast packets from being translated.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Configuring Match Conditions in NAT Rules on page 32

source-address-range

Syntax source-address-range low *minimum-value* high *maximum-value* <except>;

Hierarchy Level [edit services nat rule *rule-name* term *term-name* from]

Release Information Statement introduced in Junos OS Release 7.6.
minimum-value and ***maximum-value*** options enhanced to support IPv6 addresses in Junos OS Release 8.5.

Description Specify the source address range for rule matching.

Options ***minimum-value***—Lower boundary for the IPv4 or IPv6 address range.

maximum-value—Upper boundary for the IPv4 or IPv6 address range.

except—(Optional) Prevent the specified address range from being translated.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Configuring Match Conditions in NAT Rules on page 32

source-pool

Syntax source-pool *nat-pool-name*;

Hierarchy Level [edit services nat rule *rule-name* term *term-name* then translated]

Release Information Statement introduced before Junos OS Release 7.4.

Description Specify the source address pool for translated traffic.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Configuring Actions in NAT Rules on page 33

source-prefix

Syntax source-prefix *source-prefix*;

Hierarchy Level [edit services nat rule *rule-name* term *term-name* then translated]

Release Information Statement introduced in Junos OS Release 7.6.
source-prefix option enhanced to support IPv6 addresses in Junos OS Release 8.5.

Description Specify the source prefix for translated traffic.

Options ***source-prefix***—IPv4 or IPv6 source prefix value.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Configuring Actions in NAT Rules on page 33

source-prefix-list

Syntax source-prefix-list *list-name* <except>;

Hierarchy Level [edit services nat rule *rule-name* term *term-name* from]

Release Information Statement introduced in Junos OS Release 8.2.

Description Specify the source prefix list for rule matching. You configure the prefix list by including the **prefix-list** statement at the **[edit policy-options]** hierarchy level.

Options ***list-name***—Destination prefix list.

except—(Optional) Exclude the specified prefix list from rule matching.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Configuring Match Conditions in NAT Rules on page 32
- *Junos OS Policy Framework Configuration Guide*

syslog

Syntax syslog;

Hierarchy Level [edit services nat rule *rule-name* term *term-name* then]

Release Information Statement introduced before Junos OS Release 7.4.

Description Enable system logging. The system log information from the Multiservices PIC is passed to the kernel for logging in the **/var/log** directory.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Configuring Actions in NAT Rules on page 33

translated-port

Syntax translated-port *port id*;

Hierarchy Level [edit services nat port-forwarding *map-name*]

Release Information Statement introduced in Junos OS Release 11.4.

Description Specify the port to which all traffic will be translated.

Options ***port id***—The port number to which traffic will be translated.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- port-forwarding on page 141
- destined-port on page 131

term

Syntax

```
term term-name {
  from {
    application-sets set-name;
    applications [ application-names ];
    destination-address (address | any-unicast) <except>;
    destination-address-range low minimum-value high maximum-value <except>;
    source-address (address | any-unicast) <except>;
    source-address-range low minimum-value high maximum-value <except>;
  }
  then {
    no-translation;
    translated {
      address-pooling paired;
      destination-pool nat-pool-name;
      destination-prefix destination-prefix;
      dns-alg-pool dns-alg-pool;
      dns-alg-prefix dns-alg-prefix;
      filtering-type endpoint-independent;
      mapping-type endpoint-independent;
      source-pool nat-pool-name;
      source-prefix source-prefix;
      translation-type (basic-nat-pt | basic-nat44 | basic-nat66 | dnat-44 | dynamic-nat44
        | napt-44 | napt-66 | napt-pt | stateful-nat64 | twice-basic-nat-44 |
        twice-dynamic-nat-44 | twice-napt-44);
      }
    }
    syslog;
  }
}
```

Hierarchy Level [edit services nat rule *rule-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Define the NAT term properties.

Options *term-name*—Identifier for the term.

The remaining statements are explained separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring NAT Rules on page 30

then

Syntax

```
then {
  no-translation;
  translated {
    address-pooling paired;
    destination-pool nat-pool-name;
    destination-prefix destination-prefix;
    dns-alg-pool dns-alg-pool;
    dns-alg-prefix dns-alg-prefix;
    filtering-type endpoint-independent;
    mapping-type endpoint-independent;
    source-pool nat-pool-name;
    source-prefix source-prefix;
    translation-type (basic-nat-pt | basic-nat44 | basic-nat66 | dnat-44 | dynamic-nat44
      | napt-44 | napt-66 | napt-pt | stateful-nat64 | twice-basic-nat-44 |
      twice-dynamic-nat-44 | twice-napt-44);
    }
  }
  syslog;
}
```

Hierarchy Level [edit services nat rule *rule-name* term *term-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Define the NAT term actions.

Options The remaining statements are explained separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring NAT Rules on page 30

translated

Syntax

```
translated {
  address-pooling paired;
  destination-pool nat-pool-name;
  dns-alg-pool dns-alg-pool;
  dns-alg-prefix dns-alg-prefix;
  filtering-type endpoint-independent;
  mapping-type endpoint-independent;
  source-pool nat-pool-name;
  translation-type (basic-nat-pt | basic-nat44 | basic-nat66 | dnat-44 | dynamic-nat44 |
    napt-44 | napt-66 | napt-pt | stateful-nat64 | twice-basic-nat-44 | twice-dynamic-nat-44
    | twice-napt-44)
  }
}
```

Hierarchy Level [edit services nat rule *rule-name* term *term-name* then]

Release Information Statement introduced before Junos OS Release 7.4.

Description Define properties for translated traffic.

Options The remaining statements are explained separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Actions in NAT Rules on page 33

translation-type

Syntax translation-type (basic-nat-pt | basic-nat44 | basic-nat66 | nat-44 | deterministic-napt44 | dnat-44 | dynamic-nat44 | napt-44 | napt-66 | napt-pt | stateful-nat64 | twice-basic-nat-44 | twice-dynamic-nat-44 | twice-napt-44)

Hierarchy Level [edit services nat rule *rule-name* term *term-name* then translated]

Release Information Statement introduced before Junos OS Release 7.4.
The following options introduced in Junos OS Release 11.2, replacing all previous options:

- **basic-nat44**
- **basic-nat66**
- **basic-nat-pt**
- **dnat-44**
- **dynamic-nat44**
- **napt-44**
- **napt-66**
- **napt-pt**
- **stateful-nat64**

twice-basic-nat-44—option introduced in Junos OS Release 11.4.
twice-dynamic-nat-44—option introduced in Junos OS Release 11.4.
twice-napt-44—option introduced in Junos OS Release 11.4.
deterministic-napt44—option introduced in Junos OS Release 12.1.

Description Specify the NAT translation types.

Options

- **basic-nat44**—Translate the source address statically (IPv4 to IPv4).
- **basic-nat66**—Translate the source address statically (IPv6 to IPv6).
- **basic-nat-pt**—Translate the addresses of IPv6 hosts as they originate sessions to the IPv4 hosts in the external domain. The **basic-nat-pt** option is always implemented with DNS ALG.
- **deterministic-napt44**—Translate as napt-44, and use deterministic port block allocation for port translation.
- **dnat-44**—Translate the destination address statically (IPv4 to IPv4).
- **dynamic-nat44**—Translate only the source address by dynamically choosing the NAT address from the source address pool.
- **napt-44**—Translate the transport identifier of the IPv4 private network to a single IPv4 external address.

- **napt-66**—Translate the transport identifier of the IPv6 private network to a single IPv6 external address.
- **napt-pt**—Bind addresses in an IPv6 network with addresses in an IPv4 network and vice versa to provide transparent routing for the datagrams traversing between the address realms.
- **stateful-nat64**—Implement dynamic address and port translation for source IP addresses (IPv6-to-IPv4) and prefix removal translation for the destination IP addresses (IPv6-to-IPv4).
- **twice-basic-nat-44**—Translate the source and destination addresses statically (IPv4 to IPv4).
- **twice-dynamic-nat-44**—Translate the source address by dynamically choosing the NAT address from the source address pool. Translate the destination address statically.
- **twice-dynamic-napt-44**—Translate the transport identifier of the IPv4 private network to a single IPv4 external address. Translate the destination address statically.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Actions in NAT Rules on page 33

translation-type (Twice NAT)

Syntax

```
translation-type {
  source type;
  destination type;
}
```

Hierarchy Level [edit services nat rule *rule-name* term *term-name* then translated]

Release Information Statement introduced in Junos OS Release 8.3.

Description Specify the NAT types for Twice NAT.

Options ***type***—You must specify **destination static** and either **source dynamic** or **source static**.

NOTE: Starting with Junos OS Release 11.2, the translation types **destination static**, **source dynamic**, and **source static** are deprecated. However, these statements will continue to be supported until Junos OS Release 11.4. Therefore, you can continue to configure Twice NAT using these deprecated statements. The new statement for configuring Twice NAT will be introduced in a future release.

Usage Guidelines See "Configuring Actions in NAT Rules" on page 33.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

transport

Syntax transport [*transport-protocols*];

Hierarchy Level [edit services nat pool *nat-pool-name* pgcp]

Release Information Statement introduced in Junos OS Release 9.2.

Description Configure the BGF to select a NAT pool based on transport protocol type.

Options **[*transport-protocol*]**—One or more transport protocols.

Values: **rtp-avp**, **tcp**, **udp**

Syntax: One or more protocols. If you specify more than one protocol, you must enclose all protocols in brackets.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Session Border Control Solutions Guide Using BGF and IMSG*

use-dns-map-for-destination-translation

Syntax use-dns-map-for-destination-translation;

Hierarchy Level [edit services nat rule *rule-name* term *term-name* then translated]

Release Information Statement introduced in Junos OS Release 10.4.

Description Enable the Domain Name System (DNS) application-level gateway (ALG) address map for destination translation.

NOTE: This statement is deprecated and might be removed completely in a future release.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

CHAPTER 7

Softwire Configuration Tasks

- Configuring a DS-Lite Softwire Concentrator on page 157
- Configuring a 6rd Softwire Concentrator on page 158
- Configuring Stateful Firewall Rules for 6rd Softwire on page 158
- Configuring Softwire Rules on page 159
- Configuring Service Sets for Softwire on page 159

Configuring a DS-Lite Softwire Concentrator

To configure a DS-Lite softwire concentrator:

1. Assign a name to the DS-Lite softwire concentrator.

   [edit services softwire softwire-concentrator]
   user@host# edit ds-lite ***ds-lite-softwire-concentrator***

2. Specify the address of the softwire tunnel.

   [edit services softwire softwire-concentrator ds-lite *ds-lite-softwire-concentrator*]
   user@host# **set softwire-address *address***

3. Specify the MTU for the softwire tunnel.

   [edit services softwire softwire-concentrator ds-lite *ds-lite-softwire-concentrator*]
   user@host# **set mtu-v6 *mtu-v6***

   NOTE: This option sets the maximum transmission unit when encapsulating IPv4 packets into IPv6. If the final length is greater than the MTU, the IPv6 packet will be fragmented. This option is mandatory since it depends on other network parameters under administrator control.

4. To copy DSCP information from the IPv6 header into the decapsulated IPv4 header, include the **copy-dscp** statement.

   [edit services softwire softwire-concentrator ds-lite *ds-lite-softwire-concentrator*]
   user@host# **set copy-dscp**

5. Specify the maximum number of flows for the softwire:

   [edit services softwire softwire-concentrator ds-lite *ds-lite-softwire-concentrator*]
   user@host# **set flow-limit 1000**

Configuring a 6rd Softwire Concentrator

To configure a 6rd softwire concentrator:

1. Assign a name to the 6rd softwire concentrator.

   [edit services softwire softwire-concentrator]
   user@host# edit v6rd *v6rd-softwire-concentator*

2. Specify the address of the softwire tunnel.

   [edit services softwire softwire-concentrator v6rd *v6rd-softwire-concentator*]
   user@host# set softwire-address *address*

3. Specify the MTU for the softwire tunnel.

   [edit services softwire softwire-concentrator v6rd *v6rd-softwire-concentator*]
   user@host# set mtu-v4 *mtu-v4*

TIP: In this release there is no support for fragmentation and reassembly, therefore the MTUs on the IPv6 and IPV4 network must be properly configured by the administrator.

Configuring Stateful Firewall Rules for 6rd Softwire

You must configure a stateful firewall rule for use with 6rd softwires. The stateful firewall service is used only to direct packets to the softwire, not for firewalling purposes. The 6rd softwire service itself must be stateless. To support stateless processing, you must include an **allow** term in both directions of the stateful firewall policy.

To include a stateful firewall rule for 6rd softwire processing:

1. Assign a name to the rule.

   [edit services stateful-firewall]
   user@host# edit rule *rule-name*

2. Specify the match direction.

   [edit services stateful-firewall *rule-name*]
   user@host# set match-direction input-output

3. Assign a name for the term.

   [edit services stateful-firewall *rule-name*]
   user@host# edit term *term-name*

4. Specify that all traffic in both directions should be accepted for the softwire process.

   [edit services stateful-firewall *rule-name* term *term-name*]
   user@host# set then accept

Configuring Softwire Rules

You configure softwire rules to instruct the router how to direct traffic to the addresses specified for 6rd or DS-Lite softwire concentrators. Softwire rules do not perform any filtration of the traffic. They do not include a **from** statement, and the only option in the **then** statement is to specify the address of the 6rd or DS-Lite softwire concentrator.

You can create a softwire rule consisting of one or more terms and associate a particular 6rd or DS-Lite softwire concentrator with each term. You can include the softwire rule in service sets along with other services rules.

To configure a softwire rule:

1. Assign a name to the rule.

   [edit services softwire]
   user@host# **edit** rule *rule-name*

2. Specify the match direction.

   [edit services softwire rule *rule-name*]
   user@host# **set match-direction** (input | output)

3. Assign a name for the first term.

   [edit services softwire rule *rule-name*]
   user@host# **edit term** *term-name*

4. Associate a 6rd or DS-Lite softwire concentrator with this term.

   [edit services softwire rule *rule-name* term *term-name*]
   user@host# **set then ds-lite** *name*

   or

   user@host# **set then v6rd** *v6rd-softwire-concentator*

5. Repeat Steps 3 and 4 for as many additional terms as needed.

Configuring Service Sets for Softwire

You must include softwire rules or a softwire rule set in a service set to enable softwire processing. You must include a stateful firewall rule for DS-Lite.

To configure service sets for softwire:

1. Include a softwire rule or rule set in the service set.

   [edit services service-set *service-set-name*]
   user@host# **set** softwire-rules **rule** ***softwire-rule-name***

2. When using a 6rd softwire, include a stateful-firewall rule.

   [edit services service-set *service-set-name*]
   user@host# **set stateful-firewall-rules***softwire-rule-name*

3. You can include a NAT rule for flows originated by DS-Lite softwires.

NOTE:

Currently a NAT rule configuration is required with a DS-Lite softwire configuration when you use interface service set configurations; NAT is not required when using next-hop service set configurations. NAT processing from IPv4 to IPv6 address pools and vice versa is not currently supported. FTP, HTTP and RSTP are supported.

For further information, see Configuring Service Rules.

CHAPTER 8

Softwire Configuration Examples

- Example: Basic DS-Lite Configuration on page 161
- Example: Basic 6rd Configuration on page 166
- Example: Configuring DS-Lite and 6rd in the Same Service Set on page 169

Example: Basic DS-Lite Configuration

- Requirements on page 161
- Configuration Overview and Topology on page 161
- Configuration on page 162

Requirements

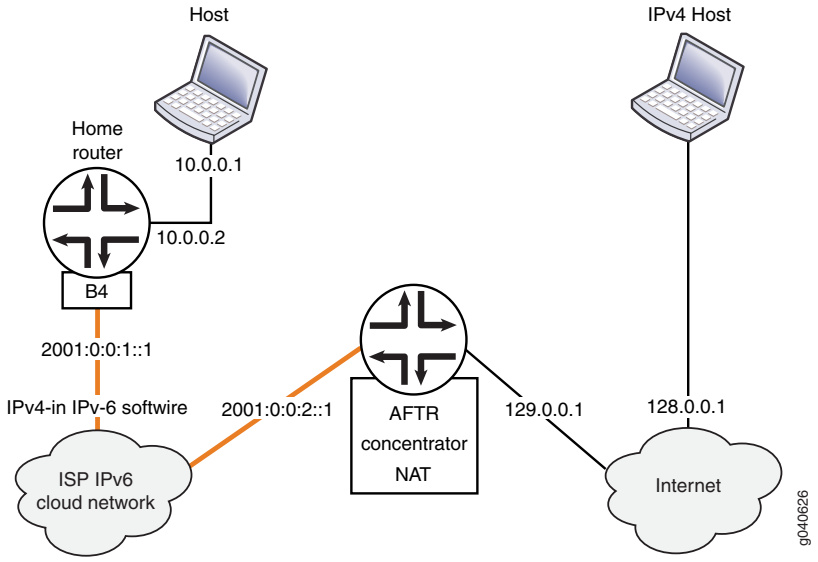
The following hardware components can perform DS-Lite:

- M Series Multiservice Edge routers with Multiservices PICs
- T Series Core routers with Multiservices PICs
- MX Series 3D Universal Edge routers with Multiservices DPCs

Configuration Overview and Topology

This example describes how configure an MX Series router with an MS-DPC as an AFTR to facilitate the flow shown in Figure 10 on page 162.

Figure 10: DS-Lite Topology



In this example, the DS-Lite softwire concentrator, or AFTR, is an MX Series router with two Gigabit interfaces and a Services DPC. The interface facing the B4 element is ge-3/1/5 and the interface facing the Internet is ge-3/1/0.

Configuration

- Chassis Configuration on page 162
- Interfaces Configuration on page 163
- Network Address and Port Translation Configuration on page 164
- Softwire Configuration on page 165
- Service Set Configuration on page 165

Chassis Configuration

Step-by-Step Procedure

To configure the service PIC (FPC 0 Slot 0) with the Layer 3 service package:

1. Enter the **edit chassis** hierarchy level.

   user@host# edit chassis

2. Configure the Layer 3 service package.

   [edit chassis]
   user@host# set fpc 0 pic 0 adaptive-services service-package layer-3

Interfaces Configuration

Step-by-Step Procedure

To configure the AFTR interfaces facing the B4 (softwire initiator) and facing the Internet:

1. Go the **[edit interfaces]** edit hierachy level for ge-3/1/0, which faces the Internet.

   ```
   host# edit interfaces ge-3/1/0
   ```

2. Define the interface.

   ```
   [edit interfaces ge-3/1/0]
   user@host# set description AFTR-Internet
   user@host# set unit 0 family inet address 128.0.0.2/24
   ```

3. Go to the **[edit interfaces]** hierachy level for ge-3/1/5, which faces the B4.

   ```
   user@host# up 1
   [edit]
   user@host# edit interfaces ge-3/1/5
   ```

4. Define the interface.

   ```
   [edit interfaces ge-3/1/5]
   user@host# set description AFTR-B4
   user@host# set unit 0 family inet
   user@host# edit unit 0 family inet6
   [edit unit 0 family inet6]
   user@host# set service input service-set sset
   user@host# set service output service-set sset
   user@host# set address 2001:0:0:2::1/48
   ```

5. Go to the **[edit interfaces]** hierarchy level for sp-0/0/0, used to host the DS-Lite AFTR.

   ```
   [edit]
   user@host# edit interfaces sp-0/0/0
   ```

6. Define the interface.

   ```
   [edit interfaces sp-0/0/0]
   user@host# set description AFTR-B4
   user@host# set unit 0 family inet
   user@host# edit unit 0 family inet6
   ```

Results

```
user@host# show interfaces ge-3/1/0
description AFTR-Internet;
unit 0 {
    family inet {
        address 128.0.0.2/24;
    }
}
```

```
user@host# show interfaces ge-3/1/5
description AFTR-B4;
unit 0 {
    family inet;
    family inet6 {
        service {
            input {
                service-set sset;
```

```
            }
            output {
                service-set sset;
            }
        }
        address 2001:0:0:2::1/48;
    }
}
```

```
user@host# show interfaces sp-0/0/0
unit 0 {
family inet;
family inet6;
}
```

Network Address and Port Translation Configuration

Step-by-Step Procedure

To configure NAPT:

1. Go to the **[edit services nat]** hierarchy level.

   ```
   user@host# edit services nat
   [edit services nat]
   ```

2. Define a NAT pool p1.

   ```
   user@host# set pool p1 address 129.0.0.1/32 port automatic
   ```

3. Define a NAT rule, beginning with the match direction.

   ```
   [edit services nat]
   user@host# set rule r1 match-direction input
   ```

4. Define a **term** for the rule, beginning with a from clause.

   ```
   [edit services nat]
   user@host# set rule r1 term t1 from source-address 10.0.0.0/16
   ```

5. Define the desired translation in a **then** clause. In this case, use dynamic source translation.

   ```
   [edit services nat]
   user@host# set rule r1 term t1 then translated source-pool p1 translation-type
     napt-44
   ```

6. (Optional) Configure logging of translation information for the rule.

   ```
   [edit services nat]
   user@host# set rule r1 term t1 then syslog
   ```

Results

```
user@host# show services nat
pool p1 {
    address 129.0.0.1/32;
    port {
        automatic;
    }
}
rule r1 {
    match-direction input;
    term t1 {
        from {
            source-address {
```

```
                10.0.0.0/16;
            }
        }
        then {
            translated {
                source-pool p1;
                translation-type {
                    napt-44;
                }
            }
            syslog;
        }
    }
```

Softwire Configuration

Step-by-Step Procedure

To configure the DS-Lite softwire concentrator and associated rules:

1. Go to the **[edit services softwire]** hierarchy.

   user@host# edit services softwire

2. Define the DS-Lite softwire concentrator.

   [edit services softwire]
   user@host# set softwire-concentrator ds-lite ds-1 softwire-address 1001::1 mtu-v6 1460

3. Define the softwire rule.

   [edit services softwire]
   user@host# set rule r1 match-direction input term t1 then ds-lite ds1.

Results

```
user@host# show services softwire
softwire-concentrator {
    ds-lite ds1 {
        softwire-address 1001::1;
        mtu-v6 1460;
    }
}
rule r1 {
    match-direction input;
    term t1 {
        then {
            ds-lite ds1;
        }
    }
}
```

Service Set Configuration

Step-by-Step Procedure

Configure a service set that includes softwire and NAT rules and specifies either interface-service or next-hop service. This example uses a next-hop service.

1. Go to the **[edit services service-set]** hierarchy level, naming the service set.

   user@host# edit services service-set sset

2. Define the NAT rule to be used for IPv4-to-IPv4 translation.

```
[edit services service-set sset]
user@host# set nat-rules r1
```

3. Define the softwire rule to define the softwire tunnel.

```
[edit services service-set sset]
user@host# set softwire-rules r1
```

4. Define the interface service,

```
[edit services service-set sset]
user@host# set interface-service service-interface sp-0/0/0.0
```

TIP: In order to avoid or minimize IPv6 fragmentation, you can configure a TCP maximum segment size (MSS) for your service set.

5. (Optional) Define a TCP MSS.

```
[edit services service-set sset]
user@host# set tcp-mss 1024
```

Results

```
user@host# show services service-set
syslog {
    host local {
        services any;
    }
}
softwire-rules r1;
nat-rules r1;
interface-service {
    service-interface sp-0/0/0;
}
}
```

Example: Basic 6rd Configuration

- Requirements on page 166
- Overview on page 167
- Configuration on page 167

Requirements

This example describes how a 6rd concentrator can be configured for a 6rd domain, D1, to provide IPv6 Internet connectivity.

The following hardware components can perform 6rd:

- M Series Multiservice Edge routers with Multiservices PICs
- T Series Core routers with Multiservices PICs
- MX Series 3D Universal Edge routers with Multiservices DPCs

Overview

This configuration example describes how to configure a basic 6rd tunneling solution.

Configuration

Chassis Configuration

Step-by-Step Procedure

To configure the chassis:

1. Define the ingress interface.

   user@host# edit interfaces ge-1/2/0

2. Configure the ingress interface logical unit and input/output service options.

   [edit interfaces ge-1/2/0]
   user@ host# set unit 0 family inet service input service-set v6rd-dom1-service-set
   user@ host# set unit 0 family inet6 service output service-set v6rd-dom1-service-set

3. Configure the address of the ingress interface.

   [edit interfaces ge-1/2/0]
   user@ host# set unit 0 family inet address 10.10.10.1/24

4. Define the egress interface.

   user@host# up 1
   [edit interfaces]
   user@host# edit ge-1/2/2

5. Define the logical unit and address for the egress interface.

   [edit interfaces ge-1/2/2]
   user@host# set unit 0 family inet6 address 3ABC::1/16

6. Define the services PIC.

   [edit interfaces ge-1/2/2]
   user@host# up 1
   [edit interfaces]
   user@host# edit sp-0/2/0

7. Configure the logical unit for the services PIC.

   [edit interfaces sp-0/2/0]
   user@host# up 1
   [edit interfaces]
   user@host# set unit 0 family inet
   user@host# set unit 0 family inet6

Softwire Concentrator, Softwire Rule, and Stateful Firewall Rule Configuration

Step-by-Step Procedure

To configure the softwire concentrator, softwire rule, and stateful firewall rule:

1. Define the 6rd softwire concentrator.

   user@host# **top**
   user@host# **edit services softwire softwire-concentrator v6rd v6rd-dom1**

2. Configure the softwire concentrator properties. Here, softwire address 30.30.30.1 is the softwire concentrator IPv4 address, 10.10.10.0/24 is the IPv4 prefix of the CE WAN side, and 3040::0/16 is the IPv6 prefix of the 6rd domain D1.

```
[edit services softwire softwire-concentrator v6rd v6rd-dom1]
user@host# set softwire-address 30.30.30.1
user@host# set ipv4-prefix 10.10.10.0/24
user@host# set v6rd-prefix 3040::0/16
user@host# set mtu-v4 9192
```

3. Define the softwire rule.

```
[edit services softwire softwire-concentrator v6rd v6rd-dom1]
user@host# up
[edit services softwire]
user@host# edit rule v6rd-dom1-r1
[edit services softwire rule v6rd-dom1-r1]
user@host# set term t1 then v6rd v6rd-dom1
```

4. Define a stateful firewall rule and properties. You must configure a stateful firewall rule that accepts all traffic in both the input and output direction in order for 6rd to work; however, this is not enforced through the CLI. This is because in IPv6, gratuitous IPv6 packets are expected (due to Anycast) and should not be dropped. The service PIC can handle reverse traffic without seeing all forward traffic. This can also happen with service PIC switchover in the middle of a session. By default, the stateful firewall on the service PIC will drop all traffic unless a rule is configured explicitly to allow it.

```
[edit services softwire softwire-concentrator v6rd v6rd-dom1]
user@host# up 2
[edit services softwire]
user@host# edit rule r1
[edit services softwire rule v6rd-dom1-r1]
user@host# set match-direction input-output
user@host# set term t1 then accept
```

Results

```
[edit services softwire]
user@router# show
softwire-concentrator {
    v6rd v6rd-dom1 {
        softwire-address 30.30.30.1;
        ipv4-prefix 10.10.10.0/24;
        v6rd-prefix 3040::0/16;
        mtu-v4 9192;
    }
}
rule v6rd-dom1-r1 {
    match-direction input;
    term t1 {
        then {
            v6rd v6rd-dom1;
        }
    }
}
```

Service Set Configuration

Step-by-Step Procedure

To configure the service set:

1. Define the service set for 6rd processing.

   ```
   user@host# top
   user@host# edit services service-set v6rd-dom1-service-set
   ```

2. Define the softwire and stateful firewall rules for the service set.

   ```
   [edit services service-set v6rd-dom1-service-set]
   user@host# set softwire-rules v6rd-dom1-r1
   user@host# set stateful-firewall-rules r1
   ```

3. Define the interface-service for the service set.

   ```
   [edit services service-set v6rd-dom1-service-set]
   user@host# set interface-service service-interface sp-3/0/0
   ```

Results

```
[edit service-set v6rd-dom1-service-set]
user@host# show
softwire-rules v6rd-dom1-r1
    interface-service {
        service-interface sp-3/0/0;
    }
```

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

- Requirements on page 169
- Overview on page 169
- Configuration on page 169

Requirements

The following hardware components can perform DS-Lite:

- M Series Multiservice Edge routers with Multiservices PICs
- T Series Core routers with Multiservices PICs
- MX Series 3D Universal Edge routers with Multiservices DPCs

Overview

This example describes a softwire solution that includes DS-Lite and 6rd in the same service set.

Configuration

Chassis Configuration

Step-by-Step Procedure

To configure the chassis:

1. Configure the ingress interface.

```
user@host# edit interfaces ge-1/2/0
[edit interfaces ge-1/2/0]
user@host# set unit 0 family inet service input service-set v6rd-dslite-service-set
user@host# set unit 0 family inet service output service-set v6rd-dslite-service-set
user@host# set unit 0 family inet address address 10.10.10.1/24
user@host# set unit 0 family inet6 service input service-set v6rd-dslite-service-set
user@host# set unit 0 family inet6 service output service-set v6rd-dslite-service-set
user@host# set unit 0 family inet6 address address address 2001::1/16
```

Here the service set is applied on the inet (IPv4) and inet6 (IPv6) families of subunit 0. Both DS-Lite IPv6 traffic and 6rd IPv4 traffic hits the service filter and is sent to the services PIC.

2. Configure the egress interface (IPv6 Internet). The IPv4 server that the DS-Lite clients are trying to reach is at 200.200.200.2/24, and the IPv6 server is at 3ABC::2/16.

```
user@host# edit interfaces ge-1/2/2
[edit interfaces ge-1/2/2]
user@host# set unit 0 family inet address 200.200.200.1/24
user@host# set unit 0 family inet6 address 3ABC::1/16
```

3. Configure the services PIC.

```
user@host# edit interfaces sp-3/0/0
[edit interfaces sp-3/0/0]
user@host# set unit 0 family inet
user@host# set unit 0 family inet6
```

Results

```
[edit interfaces]
user@host# show
ge-1/2/0 {
    unit 0 {
        family inet {
            service {
                input {
                    service-set v6rd-dslite-service-set;
                }
                output {
                    service-set v6rd-dslite-service-set;
                }
            }
            address 10.10.10.1/24;
        }
        family inet6 {
            service {
                input {
                    service-set v6rd-dslite-service-set;
                }
                output {
                    service-set v6rd-dslite-service-set;
                }
            }
            address 2001::1/16;
        }
    }
}
ge-1/2/2 {
    unit 0 {
```

```
        family inet {
            address 200.200.200.1/24;
        }
        family inet6 {
            address 3ABC::1/16;
        }
    }
}
sp-3/0/0 {
    unit 0 {
        family inet;
        family inet6;
    }
}
```

Softwire Concentrator, Softwire Rule, Stateful Firewall Rule Configuration

Step-by-Step Procedure

To configure the softwire concentrator, softwire rule, and stateful firewall rule:

1. Configure the DS-Lite and 6rd softwire concentrators.

```
user@host# edit services softwire softwire-concentrator ds-lite ds1
[edit services softwire softwire-concentrator ds-lite ds1]
user@host# set softwire-address 1001::1
user@host# mtu-v6 9192
usert@host# up 1
usert@host# edit v6rd v6rd-dom1
[edit services softwire softwire-concentrator v6rd v6rd-dom1]
user@host# set softwire-address 30.30.30.1
user@host# set ipv4-prefix 10.10.10.0/24
user@host# set v6rd-prefix 3040::0/16
user@host# set mtu-v4 9192
```

2. Configure the softwire rules.

```
user@host# edit services softwire rule v6rd-r1]
[edit services softwire rule v6rd-r1]
user@host# set match-direction input
user@host# set term t1 then v6rd v6rd-dom1
user@host# up 1
user@host# edit services softwire]
[edit services softwire]
user@host# edit rule dslite-r1
[edit services softwire rule dslite-r1]
user@host# set term dslite-t1 then ds-lite ds1
```

The following routes are added by the services PIC daemon on the Routing Engine:

```
user@router# run show route 30.30.30.1

inet.0: 43 destinations, 46 routes (42 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

30.30.30.1/32      *[Static/786432] 00:24:11
                       Service to v6rd-dslite-service-set

[edit]
user@router# run show route 3040::0/16

inet6.0: 23 destinations, 33 routes (23 active, 0 holddown, 0 hidden)
```

```
+ = Active Route, - = Last Active, * = Both

3040::/16          *[Static/786432] 00:24:39
                      Service to v6rd-dslite-service-set
```

```
user@router# run show route 1001::1

inet6.0: 33 destinations, 43 routes (33 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1001::1/128        *[Static/1] 1w2d 22:05:41
                      Service to v6rd-dslite-service-set
```

3. Configure a stateful firewall rule.

```
user@host# edit services stateful-firewall rule r1
[edit services stateful-firewall rule r1]
user@host# set match-direction input-output
user@host# set term t1 then accept
```

```
[edit services stateful-firewall]
  rule r1 {
    match-direction input-output;
    term t1 {
      then {
        accept;
      }
    }
  }
```

Results

```
[edit services softwire]
user@host# show
softwire-concentrator {
    ds-lite ds1 {
        softwire-address 1001::1;
        mtu-v6 9192;
    }
    v6rd v6rd-dom1 {
        softwire-address 30.30.30.1;
        ipv4-prefix 10.10.10.0/24;
        v6rd-prefix 3040::0/16;
        mtu-v4 9192;
    }
}
rule v6rd-r1 {
    match-direction input;
    term t1 {
        then {
            v6rd v6rd-dom1;
        }
    }
}
rule dslite-r1 {
    match-direction input;
    term dslite-t1 {
        then {
            ds-lite ds1;
        }
```

```
    }
}

[edit services stateful-firewall]
user@host# show
rule r1 {
    match-direction input-output;
    term t1 {
        then {
            accept;
        }
    }
}
```

NAT Configuration for DS-Lite

Step-by-Step Procedure

To configure NAT for DS-Lite:

1. Configure a NAT pool for DS-Lite.

```
user@host# edit services nat pool dslite-pool
[edit services nat pool dslite-pool]
user@host# set address-range low 33.33.33.1 high 33.33.33.32
user@host# set port automatic
```

2. Configure a NAT rule.

```
user@host# up 1
[edit services nat rule dslite-nat-r1]
user@host# set match-direction input
user@host# set term dslite-nat-t1 from source-address 20.20.0.0/16 then translated
  translation-type napt-44
```

Results

```
[edit services nat]
user@host# show
pool dslite-pool {
    address-range low 33.33.33.1 high 33.33.33.32;
    port {
        automatic;
    }
}
rule dslite-nat-r1 {
    match-direction input;
    term dslite-nat-t1 {
        from {
            source-address {
                20.20.0.0/16;
            }
        }
        then {
            translated {
                source-pool dslite-pool;
                translation-type {
                    source dynamic;
                }
            }
        }
```

```
    }
}
```

Because of this NAT rule, the following NAT routes are installed for the reverse DS-Lite traffic:

```
user@router# run show route 33.33.33.0/24
inet.0: 48 destinations, 52 routes (47 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

33.33.33.1/32      *[Static/1] 1w2d 23:08:38
                      Service to v6rd-dslite-service-set
33.33.33.2/31      *[Static/1] 1w2d 23:08:38
                      Service to v6rd-dslite-service-set
33.33.33.4/30      *[Static/1] 1w2d 23:08:38
                      Service to v6rd-dslite-service-set
33.33.33.8/29      *[Static/1] 1w2d 23:08:38
                      Service to v6rd-dslite-service-set
33.33.33.16/28     *[Static/1] 1w2d 23:08:38
                      Service to v6rd-dslite-service-set
33.33.33.32/32     *[Static/1] 1w2d 23:08:38
                      Service to v6rd-dslite-service-set
```

The NAT rule triggers address translation for the traffic coming from 20.20.0.0/16 to public address range 33.33.33.1 to 33.33.33.32.

Service Set Configuration

Step-by-Step Procedure

This service set has a stateful firewall rule and 6rd rule for 6rd service. The service set also includes a softwire rule for DS-Lite and a NAT rule to perform address translation for all DS-Lite traffic. The NAT rule performs NAPT translation in the forward direction on the source address and port of the DS-Lite traffic.

To configure the service set:

1. Define the service set.

   ```
   user@host# edit services service-set v6rd-dslite-service-set
   ```

2. Configure the service set rules.

   ```
   [edit services service-set v6rd-dslite-service-set]
   user@host# set softwire-rules dslite-r1
   user@host# set stateful-firewall-rules r1
   user@host# set nat-rules dslite-nat-r1
   ```

3. Configure the service set interface-service.

   ```
   [edit services service-set v6rd-dslite-service-set]
   user@host# set interface-service service-interface sp-3/0/0
   ```

Results

```
[edit services service-set]
user@host# show
v6rd-dslite-service-set {
    softwire-rules v6rd-r1;
    softwire-rules dslite-r1;
    stateful-firewall-rules r1;
    nat-rules dslite-nat-r1;
    interface-service {
```

```
    service-interface sp-3/0/0;
}
```


CHAPTER 9

6to4 Configuration

- Configuring a 6to4 Provider-Managed Tunnel on page 177

Configuring a 6to4 Provider-Managed Tunnel

When configuring a 6to4 provider-managed tunnel (PMT), replace the Anycast destination with the address of a managed relay in the provider network.

To configure a 6to4 PMT:

1. Configure the ingress interface for 6to4 traffic. Include the name of the service set that identifies the rules for input and output service on this interface.

   [edit interfaces ge-0/2/1]
   user@host# set unit *logical-unit-number* family *family* service input *service-set-name*
   user@host# set unit *logical-unit-number* family *family* service output *service-set-name*
   user@host# set unit *logical-unit-number* family *family* address *addres*

   For example:

   [edit interfaces ge-0/2/1]
   user@host# set unit 0 family inet service input service-set v6to4-pmt
   user@host# set unit 0 family inet service output service-set v6to4-pmt
   user@host# set unit 0 family inet address 130.130.130.1/24
2. Configure the egress interface.

   [edit interfaces ge-0/2/2]
   user@host# set unit *logical-unit-number* family *family* address *address*

   For example:

   [edit interfaces ge-0/2/2]
   user@host# set unit 0 family inet6 address 4ABC::1/16
3. Configure the service interface that contains the rules for processing incoming traffic. Include a syslog option and associate a logical unit.

   [edit interfaces sp-2/0/0]
   user@host# edit services-options syslog host *host-name* services any
   user@host# edit unit *logical-unit-number* family *family*
   user@host# edit unit 0 family *family*

   For example:

   [edit interfaces sp-2/0/0]

```
user@host# set services-options syslog host local services any
user@host# set unit 0 family inet
user@host# set unit 0 family inet6
```

4. Configure the softwire concentrator and softwire rule for 6to4. In the Junos OS, 6to4 PMT configuration uses the same options as 6rd.

   ```
   [edit services softwire softwire-concentrator v6rd v6to4]
   user@host# set softwire-address softwire-addres
   user@host# set ipv4-prefix ipv4-prefix
   user@host# set v6rd-prefix v6rd-prefix
   user@host# set mtu-v4 mtu-v4
   ```

   For example:

   ```
   [edit services softwire softwire-concentrator v6rd v6to4]
   user@host# set softwire-address 192.88.99.1
   user@host# set ipv4-prefix 130.130.130.2/32
   user@host# set v6rd-prefix 2002::0/16
   user@host# set mtu-v4 9192
   ```

5. Define the softwire rule that will process traffic on the ingress interface.

   ```
   [edit services softwire rule v6to4-r1]
   user@host# set match-direction input
   user@host# set term term-name then v6rd softwire-concentrator
   ```

   For example:

   ```
   [edit services softwire rule v6to4-r1]
   user@host# set match-direction input
   user@host# set term t1 then v6rd v6to4
   ```

6. Define a stateful firewall rule that will accept all incoming traffic on the ingress interface.

   ```
   [edit services stateful-firewall rule sfw-r1]
   user@host# set match-direction direction
   user@host# set term term-name then accept
   user@host# set term term-name then syslog
   ```

   For example:

   ```
   [edit services stateful-firewall rule sfw-r1]
   user@host# set match-direction input-output
   user@host# set term t1 then accept
   user@host# set term t1 then syslog
   ```

7. Define the NAT pool to be used for IPv6 NAT translation. This pool supports translation of the Anycast 6to4 relay addresses to addresses at the provider-managed relay.

   ```
   [edit services nat pool v6to4-pmt]
   user@host# set address address
   user@host# port automatic
   ```

   For example:

   ```
   [edit services nat pool v6to4-pmt]
   user@host# set address 3ABC::1/128
   user@host# set port automatic
   ```

8. Define the NAT rule for translation.

```
[edit services nat rule rule-name]
user@host# set match-direction input
user@host# set term term-name then translated source-pool pool-name
user@host# set term t1 then translated translation-type translation-type
```

For example:

```
[edit services nat rule v6to4-pmt-r1]
user@host# set match-direction input
user@host# set term t1 then translated source-pool v6to4-pmt
user@host# set term t1 then translated translation-type napt-66
```

9. Define the service set that specifies the softwire rule and NAT rule.

```
[edit services service-set v6to4-pmt]
user@host# set softwire-rules rule-name
user@host# set stateful-firewall-rules rule-name
user@host# set nat-rules rule-name
user@host# set interface-service service-interface interface-name
```

For example:

```
[edit services service-set v6to4-pmt]
user@host# set softwire-rules v6to4-r1
user@host# set stateful-firewall-rules sfw-r1
user@host# set nat-rules v6to4-pmt-r1
user@host# set interface-service service-interface sp-2/0/0
```


CHAPTER 10

Softwire Configuration Statements

ds-lite

Syntax

```
ds-lite ds-lite-softwire-concentrator{
    auto-update-mtu;
    copy-dscp;
    flow-limit flow-limit;
    mtu-v6 mtu-v6;
    softwire-address softwire-address;
  }
}
```

Hierarchy Level [edit services softwire softwire-concentrator]

Release Information
Statement introduced in Junos OS Release 10.4.
auto-update-mtu option introduced in Junos OS Release 10.4.
copy-dscp option introduced in Junos OS Release 11.2.
mtu-v6 option introduced in Junos OS Release 10.4.
softwire-address option introduced in Junos OS Release 10.4.

Description Configure settings for a DS-Lite concentrator used to process IPv4 packets encapsulated in IPv6.

Options

ds-lite-softwire-concentrator—Name applied to a DS-Lite softwire concentrator.

auto-update-mtu—This option is not currently supported.

copy-dscp—Copy DSCP information to IPv4 headers during decapsultation.

flow-limit—Maximum number of IPv4 flows per softwire (0 through 16384).

mtu-v6—Maximum transmission unit (MTU), in bytes (0 through 9192), for encapsulating IPv4 packets into IPv6. If the final length is greater than the configured value, the IPv6 packet is fragmented.

softwire-address—Address of the DS-Lite softwire concentrator.

Required Privilege Level
interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Softwire Configuration Guidelines
- Configuring a DS-Lite Softwire Concentrator on page 157

rule (Softwire)

Syntax

```
rule rule-name {
  match-direction (input | output);
  term term-name {
    then {
      (ds-lite ds-lite-softwire-concentrator | v6rd v6rd-softwire-concentrator);
    }
  }
}
```

Hierarchy Level [edit services softwire],
[edit services softwire rule-set *rule-set-name*]

Release Information Statement introduced in Junos OS Release 10.4.

Description Configure a rule to apply a softwire concentrator for a flow.

Options ***rule-name***—Identifier for the collection of terms that constitute this rule.

input—Apply the rule match on the input side of the interface.

output—Apply the rule match on the output side of the interface.

The remaining statements are explained separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Configuring Softwire Rules on page 159

rule-set (Softwire)

Syntax

```
rule-set rule-set-name {
  rule rule-name;
}
```

Hierarchy Level [edit services softwire]

Release Information Statement introduced in Junos OS Release 10.4.

Description Specify the rule set the router uses when applying this service.

Options ***rule-set-name***—Identifier for the collection of rules that constitute this rule set.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Configuring Softwire Rules on page 159

softwire-concentrator

Syntax

```
softtwire-concentrator {
  ds-lite ds-lite-softwire-concentrator {
    auto-update-mtu;
    flow-limit flow-limit;
    mtu-v6 mtu-v6;
    softwire-address address;
  }
  v6rd v6rd-softwire-concentator {
    ipv4-prefix ipv4-prefix;
    v6rd-prefix ipv6-prefix;
    mtu-v4 mtu-v4;
  }
}
```

Hierarchy Level [edit services softwire]

Release Information Statement introduced in Junos OS Release 10.4.

Description Configure settings for a softwire concentrator.

Options The statements are explained separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Softwire Configuration Guidelines

softwire-rules

Syntax (softwire-rule *rule-name* | softwire-rule-sets *rule-set-name*);

Hierarchy Level [edit services service-set *service-set-name*]

Release Information Statement introduced in Junos OS Release 10.4.

Description Specify the DS-Lite or 6rd rules or rule set included in this service set. You can configure multiple rules; however, you can only configure one rule set for each service set.

Options ***rule-name***—Identifier for the collection of terms that constitute this rule.

rule-set-name—Identifier for the set of rules to be included.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Service Rules

v6rd

Syntax

```
v6rd v6rd-softwire-concentator {
  ipv4-prefix ipv4-prefix;
  v6rd-prefix ipv6-prefix;
  mtu-v4 mtu-v4;
  softwire-address ipv4-address;
}
```

Hierarchy Level [edit services softwire softwire-concentrator]

Release Information Statement introduced in Junos OS Release 10.4.

Description Configure settings for a 6rd concentrator used to process IPv6 packets encapsulated in IPv4 packets.

Options

ipv4-prefix—IPv4 prefix of the customer edge (CE) network

ipv6-prefix—IPv6 prefix of the 6rd domain.

mtu-v4— Maximum transmission unit (MTU), in bytes (576 through 9192), for IPv6 packets enacapsulated into IPv4. If the final length is greater than the configured value, the IPv4 packet will be dropped.

address—IPv4 address of a softwire concentrator. This is an IPv4 address independent of any interface and on a different prefix.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Softwire Configuration Guidelines

PART 3

Administration

- Monitoring CGN and Softwire Tunnels on page 189
- Logging on page 193
- High-Availability and Load Balancing on page 195
- Network Address Translation Operational Mode Commands on page 201

CHAPTER 11

Monitoring CGN and Softwire Tunnels

- Monitoring CGN, Stateful Firewall, and Softwire Flows on page 189
- Monitoring Stateful Firewall Conversations on page 190
- Monitoring Global Stateful Firewall Statistics on page 190
- Monitoring NAT Pool Usage on page 190
- Monitoring Softwire Statistics on page 191
- Ping and Traceroute for DS-Lite on page 192

Monitoring CGN, Stateful Firewall, and Softwire Flows

Purpose Use the following commands to check the creation of the softwires, pre-NAT flows, and post-NAT flows. Output can be filtered using more specific fields such as AFTR or B4 address or both for DS-Lite, and softwire-concentrator or softwire-initiator or both for 6rd.

- show services stateful-firewall flows
- show services softwire flows

Action

```
user@host# show services stateful-firewall flows
Interface: sp-0/1/0, Service set: dslite-svc-set2
Flow                                                State    Dir       Frm count
TCP      200.200.200.2:80    ->     44.44.44.1:1025  Forward  O         219942
    NAT dest        44.44.44.1:1025    ->       20.20.1.4:1025
    Softwire           2001::2         ->         1001::1
TCP          20.20.1.2:1025  ->  200.200.200.2:80    Forward  I         110244
    NAT source       20.20.1.2:1025    ->      44.44.44.1:1024
    Softwire           2001::2         ->         1001::1
TCP      200.200.200.2:80    ->     44.44.44.1:1024  Forward  O         219140
    NAT dest        44.44.44.1:1024    ->       20.20.1.2:1025
    Softwire           2001::2         ->         1001::1
DS-LITE         2001::2      ->        1001::1       Forward  I         988729
TCP      200.200.200.2:80    ->     44.44.44.1:1026  Forward  O         218906
    NAT dest        44.44.44.1:1026    ->       20.20.1.3:1025
    Softwire           2001::2         ->         1001::1
TCP          20.20.1.3:1025  ->  200.200.200.2:80    Forward  I         110303
    NAT source       20.20.1.3:1025    ->      44.44.44.1:1026
    Softwire           2001::2         ->         1001::1
TCP          20.20.1.4:1025  ->  200.200.200.2:80    Forward  I         110944
```

```
NAT source       20.20.1.4:1025    ->      44.44.44.1:1025
Softwire           2001::2         ->         1001::1
```

Related Documentation

- NAT Objects MIB in SNMP MIBs and Traps Reference
- Network Address Translation Resources–Monitoring MIB in SNMP MIBs and Traps Reference
- NAT Trap Definitions in SNMP MIBs and Traps Reference

Monitoring Stateful Firewall Conversations

Purpose Use the **show services stateful-firewall conversations** command to show conversations, or collections of related flows.

Action

```
user@host# show services stateful-firewall conversations
Interface: sp-0/0/0, Service set: sset
Conversation: ALG protocol: tcp
Number of initiators: 1, Number of responders: 1
Flow State Dir Frm
count
TCP 10.0.0.1:1025 -> 128.0.0.1:80 Forward I 372755
NAT source 10.0.0.1:1025 -> 129.0.0.1:1024
Softwire 2001:0:0:1::1 -> 1001::1
TCP 128.0.0.1:80 -> 129.0.0.1:1024 Forward O 794083
NAT dest 129.0.0.1:1024 -> 10.0.0.1:1025
Softwire 2001:0:0:1::1 -> 1001::1
```

Monitoring Global Stateful Firewall Statistics

Purpose Use the **show services stateful-firewall statistics** command to observe statistics for service sets containing softwire rules.

Action

```
user@host# show services stateful-firewall statistics
Interface Service set Accept Discard Reject Errors
sp-0/0/0 dslite-svc-set2 118991296 0 0 0
sp-0/1/0 dslite-svc-set1 237615050 0 0 0
```

Monitoring NAT Pool Usage

Purpose Use the **show services nat pool detail** command to find global NAT statistics related to pool usage. This command is frequently used in conjunction with the **show services stateful-firewall statistics** command.

Action

```
user@host# show services nat pool detail

Interface: ms-1/0/0, Service set: s1
  NAT pool: dest-pool, Translation type: DNAT-44
    Address range: 10.10.10.2-10.10.10.2
  NAT pool: napt-pool, Translation type: NAPT-44
    Address range: 50.50.50.1-50.50.50.254
    Port range: 1024-63487, Ports in use: 0, Out of port errors: 0, Max ports
used: 0
  NAT pool: source-dynamic-pool, Translation type: DYNAMIC NAT44
    Address range: 40.40.40.1-40.40.40.254
```

```
    Out of address errors: 0, Addresses in use: 0
  NAT pool: source-static-pool, Translation type: BASIC NAT44
    Address range: 30.30.30.1-30.30.30.254
```

Related Documentation

- NAT Objects MIB in SNMP MIBs and Traps Reference
- Network Address Translation Resources–Monitoring MIB in SNMP MIBs and Traps Reference
- NAT Trap Definitions in SNMP MIBs and Traps Reference

Monitoring Softwire Statistics

Purpose You can review softwire global statistics by using the show services softwire or show services softwire statistics command.

Action

```
user@host# show services softwire
Interface: sp-0/0/0, Service set: sset
Softwire Direction Flow count
2001:0:0:1::1 -> 1001::1 I 3

user@host# show services softwire statistics
DS-Lite Statistics:
Service PIC Name: :sp-0/0/0
Statistics
----------
Softwires Created :2
Softwires Deleted :1
Softwires Flows Created :2
Softwires Flows Deleted :1
Slow Path Packets Processed :2
Fast Path Packets Processed :274240
Fast Path Packets Encapsulated :583337
Rule Match Failed :0
Rule Match Succeeded :2
IPv6 Packets Fragmented :0
Transient Errors
----------------
Flow Creation Failed - Retry :0
Slow Path Failed - Retry :0
Errors
------
Softwire Creation Failed :0
Flow Creation Failed :0
Slow Path Failed :0
Packet not IPv4-in-IPv6 :0
IPv6 Fragmentation Error :0
Slow Path Failed - IPv6 Next Header Offset :0
Decapsulated Packet not IPv4 :0
Fast Path Failed - IPv6 Next Header Offset :0
No Softwire ID :0
No Flow Extension :0
Flow Limit Exceeded :0
6rd Statistics:
Service PIC Name :sp-0/0/0
Statistics
----------
Softwires Created :0
```

```
Softwires Deleted :0
Softwires Flows Created :0
Softwires Flows Deleted :0
Slow Path Packets Processed :0
Fast Path Packets Processed :0
Fast Path Packets Encapsulated :0
Rule Match Failed :0
Rule Match Succeeded :0
Transient Errors
----------------
Flow Creation Failed - Retry :0
Slow Path Failed - Retry :0
Errors
------
Softwire Creation Failed :0
Flow Creation Failed :0
Slow Path Failed :0
Packet not IPv6-in-IPv4 :0
Slow Path Failed - IPv6 Next Header Offset :0
Decapsulated Packet not IPv6 :0
Encapsulation Failed - No packet memory :0
No Softwire ID :0
No Flow Extension :0
ICMPv4 Dropped Packets :0
```

Ping and Traceroute for DS-Lite

With Junos OS Release 11.4, you can use the **ping** and **traceroute** commands to determine the status of the DS-Lite softwire tunnels:

- IPv6 ping—The softwire address endpoint on the DS-Lite softwire terminator (AFTR) is usually configured only at the **[edit services softwire]** hierarchy level; it need not be hosted on any interface. Previous releases of the Junos OS software did not provide replies to pings to the IPv6 softwire address when the AFTR was not configured on a specific interface or loopback. An IPv6 ping enables the softwire initiator (B4) to verify the softwire address of the AFTR before creating a tunnel.
- IPv4 ping—A special IPv4 address, 192.0.0.1, is reserved for the AFTR. Previous releases of the Junos OS did not respond to any pings sent to this address. A B4 and other IPv4 nodes can now ping to this address to determine whether the DS-Lite tunnel is working.
- Traceroute—The AFTR now generates and forwards traceroute packets over the DS-Lite tunnel.

NOTE: No additional CLI configuration is necessary to use the new functionality.

CHAPTER 12

Logging

- Log Generation on page 193

Log Generation

The Multiservices PIC uses the system logging protocol to generate session logging. System log messages can be sent directly from the services PIC to an external system logging server. This requires that the services PIC interface have an IP address and appropriate system logging options configured, as in this example:

```
[edit interfaces sp-5/0/0]
services-options {
  syslog {
    host 130.0.0.1 {
      services any;
    }
  }
}
  unit 0 {
    family inet {
      address 150.0.0.1/32;
    }
  }
```

Log Format

For each session, three logs are generated. The three logs allow correlation of start and end times for each session.

```
Jun 28 15:29:20 cypher (FPC Slot 5, PIC Slot 0) {sset2}[FWNAT]:
ASP_SFW_CREATE_ACCEPT_FLOW: proto 6 (TCP) application: any,
ge-1/3/5.0:10.0.0.1:8856 -> 128.0.0.2:80, creating forward or watch flow ; source
 address and port translate to 129.0.0.1:1028
Jun 28 15:29:23 cypher (FPC Slot 5, PIC Slot 0)
{sset2}[FWNAT]:ASP_NAT_POOL_RELEASE: natpool release 129.0.0.1:1028[1]
Jun 28 15:29:23 cypher (FPC Slot 5, PIC Slot 0) {sset2}[FWNAT]:
ASP_SFW_DELETE_FLOW: proto 6 (TCP) application: any, (null)(null)10.0.0.1:8856
-> 128.0.0.2:80, deleting forward or watch flow ; source address and port translate
 to 129.0.0.1:1028
```

System Log Throttling—You can limit logging with the **message-rate-limit** command.

Related Documentation

- message-rate-limit
- Configuring System Logging for Service Sets

CHAPTER 13

High-Availability and Load Balancing

- High Availability for Softwires Using Services PIC Redundancy on page 195
- Load Balancing a 6rd Domain Across Multiple Services PICs on page 195
- Example: Load Balancing a 6rd Domain Across Multiple Services PICs on page 195
- Configuring High Availability for 6rd Using 6rd Anycast on page 200

High Availability for Softwires Using Services PIC Redundancy

You can provide stateful, warm-standby backup by using a redundant services PIC, or rsp, interface in the service sets where you define your softwire rules. The rsp interface definition identifies two services PICs: a primary and secondary. When the primary PIC goes down, the service switches to the secondary PIC automatically.

For more information on how to configure service PIC redundancy, see Configuring AS or Multiservices PIC Redundancy.

Load Balancing a 6rd Domain Across Multiple Services PICs

The 6rd domain is an IPv6 network, which can potentially be very large. A single PIC, or network processing unit (NPU) on a Multiservices DPC, might not be able to handle all the traffic for the 6rd domain. To alleviate load problems, you can load-balance the 6rd domain traffic across multiple PICs. To do so, assign the same softwire rule to different services sets that use different interfaces. Configure explicit routes and equal-cost multipath (ECMP) to load-balance the 6rd traffic.

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

- Hardware and Software Requirements on page 196
- Overview on page 196
- Configuration on page 196

Hardware and Software Requirements

This example requires the following hardware:

- An MX Series 3D Universal Edge router with a services DPC with two available NPUs or an M Series Multiservice Edge router with two services PICs available for 6rd softwire concentrator processing.
- A domain name server (DNS).

This example uses the following software:

- Junos OS Release 11.4 or higher.

Overview

Because of anticipated volume, a provider needs to balance 6rd softwire traffic between two services PICs.

Configuration

- Chassis Configuration on page 196
- Softwire Concentrator and Softwire Rule Configuration on page 197
- Stateful Firewall Configuration on page 197
- Service Set Configuration on page 198
- Load-Balancing Configuration on page 198

Chassis Configuration

Step-by-Step Procedure

To configure the chassis:

1. Define the ingress interface and its properties.

   ```
   user@host# edit interfaces ge-1/2/0
   user@host# set unit 0 family inet address 10.10.10.1/16
   ```

2. Define the egress interface and its properties. In this example, the IPv6 clients try to reach the IPv6 server at 3abc::2/16.

3. Define the services PICs for selection as softwire concentrators by the load-balancing process. This configuration uses two PICs/NPUs: sp-3/0/0 and sp-3/1/0. A next-hop style service set is configured (shown in the next section).

   ```
   user@host# edit interfaces sp-3/0/0
   [edit interfaces ge-3/0/0]
   user@host# set services-options syslog host local services any
   user@host# set unit 0 family inet
   user@host# set unit 0 family inet6
   user@host# set unit 1 family inet service-domain inside
   user@host# set unit 1 family inet service-domain outside
   user@host# set unit 2 family inet service-domain inside
   user@host# set unit 2 family inet service-domain outside
   user@host# up 1
   [edit]
   ```

```
user@host# edit interfaces sp-3/1/0
[edit interfaces sp-3/1/0]
user@host# set services-options syslog host local services any
user@host# set unit 0 family inet
user@host# set unit 0 family inet6
user@host# set unit 1 family inet service-domain inside
user@host# set unit 1 family inet service-domain outside
user@host# set unit 2 family inet service-domain inside
user@host# set unit 2 family inet service-domain outside
```

Softwire Concentrator and Softwire Rule Configuration

Step-by-Step Procedure

The softwire configuration is straightforward. In this example, the 6rd domain prefix is 3040::0/16, the 6rd softwire concentrator IPv4 address is 30.30.30.1, and the customer IPv4 network is 10.10.0.0/16. In the customer premises equipment (CPE) network, all customer edge (CE) devices have addresses that belong to the 10.10.0.0/16 network. To configure the softwire:

1. Go to the **[edit services softwire]** hierarchy level.

   ```
   user@host# edit services softwire
   ```

2. Configure IPv6 multicast.

   ```
   [edit services softwire]
   user@host# set ipv6-multicast-interfaces all
   ```

3. Go to the softtwire concentrator v6rd hierarchy level and name the softwire concentrator **shenick01-rd1**.

   ```
   [edit services softwire]
   user@host# edit softwire-concentrator v6rd shenick01-rd1
   ```

4. Configure the softwire concentrator properties.

   ```
   [edit services softwire softwire-concentrator v6rdshenick01-rd1 ]
   user@host# set softwire-address 30.30.30.1
   user@host# set ipv4-prefix 10.10.0.0/16
   user@host# set v6rd-prefix 3040::/16
   user@host# set mtu-v4 9192
   ```

5. Configure a softwire rule for incoming 6rd traffic.

   ```
   [edit services softwire softwire-concentrator v6rd shenick01-rd1 ]
   user@host# up 1
   [edit services softwire ]
   user@host# edit rule shenick01-r1
   [edit services softwire rule shenick01-r1]
   user@host# set match-direction input
   user@host# set term t1 then v6rd shenick01-rd1
   ```

Stateful Firewall Configuration

Step-by-Step Procedure

1. Go to the stateful firewall hierarchy level and define a rule.

   ```
   user@host# edit services stateful-firewall rule r1
   ```

2. Set the match direction.

```
[edit services stateful-firewall rule r1]
user@host# set match-direction input-output
```

3. Configure a term that accepts all traffic.

```
[edit services stateful-firewall rule r1]
user@host# set term t1 then accept
```

Service Set Configuration

Step-by-Step Procedure

This configuration provides two service sets, each pointing to a different network processing unit (NPU). Both service sets use the same stateful firewall and softwire rules. Because they use the same softwire rule, they refer to same 6rd softwire concentrator. This results in the software concentrator being hosted on both the NPUs.

To configure the service set:

1. Define a service set for the first NPU.

```
user@host# edit services service-set v6rd-sset1
```

2. Configure the softwire and stateful firewall rules for the first NPU.

```
[edit services service-set v6rd-sset1]
user@host# set softwire-rules shenick01-r1
user@host# set stateful-firewall-rules r1
```

3. Configure the inside and outside interfaces for the next-hop service.

```
[edit services service-set v6rd-sset1]
user@host# set next-hop-service inside-service-interface sp-3/0/0.1
user@host# set next-hop-service outside-service-interface sp-3/0/0.2
```

4. Define a service set for the second NPU.

```
user@host# edit services service-set v6rd-sset2
```

5. Configure the softwire and stateful firewall rules for the second NPU.

```
[edit services service-set v6rd-sset2]
user@host# set softwire-rules shenick01-r1
user@host# set stateful-firewall-rules r1
```

6. Configure the inside and outside interfaces for the next-hop service.

```
[edit services service-set v6rd-sset1]
user@host# set next-hop-service inside-service-interface sp-3/1/0.1
user@host# set next-hop-service outside-service-interface sp-3/1/0.2
```

Load-Balancing Configuration

Step-by-Step Procedure

1. Following the instructions below to complete the configuration

   Configure explicit routes and ECMP to load-balance the 6rd traffic. Explicit routes are configured for both the 6rd concentrator IPv4 address and the 6rd domain prefix, so that they point to both NPUs. In addition to these routes, the service PIC daemon (spd) also adds default routes to these addresses pointing to the NPUs. However, the routes added by the spd use different metrics, which are computed based on the FPC, PIC, slot numbers, and the subunit of the servics PIC if used in the service

set configuration. The static routes configured in this sample configuration will have metrics of 5 and therefore a higher preference than the spd-added routes.

```
rib inet6.0 {
  static {
    route 3040::0/16 next-hop [ sp-3/0/0.2 sp-3/1/0.2 ];
  }
}
  static {
    route 30.30.30.1/32 next-hop [ sp-3/0/0.1 sp-3/1/0.1 ];
  }
```

The explicitly configured routes are as follows:

```
root@router# run show route 30.30.30.1

inet.0: 37 destinations, 40 routes (36 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

30.30.30.1/32      *[Static/5] 00:00:10
                    > via sp-3/0/0.1
                      via sp-3/1/0.1
                    [Static/786433] 00:23:03
                    > via sp-3/0/0.1
                    [Static/851969] 00:00:09
                    > via sp-3/1/0.1

root@router# run show route 3040::/16

inet6.0: 20 destinations, 33 routes (20 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

3040::/16          *[Static/5] 00:00:15
                      via sp-3/0/0.2
                    > via sp-3/1/0.2
                    [Static/786434] 00:23:08
                    > via sp-3/0/0.2
                    [Static/851970] 00:00:14
                    > via sp-3/1/0.2
```

BEST PRACTICE: The spd-installed routes have higher metric values (hence a low preference) and the metrics are different. If the metrics are different and ECMP is not enabled, even though multiple routes exist for the same destination, only one of the routes is picked up all the time (based on the metric). For ECMP you must configure equal-cost routes, and hence a manual configuration of routes is needed as shown above.

Configuring ECMP

Under the forwarding options, indicate to the router how to load-balance traffic by configuring the hash key, as shown:

```
user@host#  show forwarding-options

hash-key {
    family inet {  <== IPv4 traffic from CEs uses this
```

```
            layer-3 {
                destination-address;
                source-address;
            }
        }
        family inet6 {  <== IPv6 traffic from Internet uses this
            layer-3 {
                destination-address;
                source-address;
            }
        }
    }
```

TIP: Both IPv4 and IPv6 hash keys must be configured. The IPv4 hash key is used to distribute the traffic coming from CPE devices to the 6rd branch relay. The IPv6 hash key is used to distribute the traffic coming from the IPv6 Internet to the 6rd domain. Note that since the hash in the forward and reverse direction is on different families, different flows from the same session can end up on different NPUs. However, since 6rd processing is stateless (as far as mapping IPv6 packets to softwires is concerned) this should not be a problem.

Related Documentation

- Configuring and Verifying Load Balancing
- Configuring the IPv4 Address Family to Load-Balance LSP Traffic

Configuring High Availability for 6rd Using 6rd Anycast

You configure 6rd Anycast by defining two service sets that use the same softwire rule in both service sets, just as you do when you configure load balancing for 6rd. However, you do not configure ECMP, and as a result, the services PIC daemon (spd) installs two routes *each* for the softwire concentrator address and 6rd domain pointing to each service interface. The forwarding plane can select any route based on the priority, which is computed when the spd installs the routes. The priority is computed based on the FPC, PIC, slot numbers, and subunit number used on the sp- interface. *Only one PIC is used* based on the route priority, and that PIC gets all of the 6rd traffic. If the PIC goes down. the route pointing to it is also deleted and the forwarding plane automatically selects the alternate available PIC.

6rd Anycast is completely stateless. The spd installs the route and doesn't run any state machine for the PIC. Because the routes are pre-installed and service sets are already on the PIC, there is no service delay if a failover occurs.

CHAPTER 14

Network Address Translation Operational Mode Commands

clear services inline nat pool

Syntax clear services inline nat pool *pool-name*

Release Information Command introduced in Junos OS Release 11.4.

Description Clear global inline nat statistics

Options **pool-name**—Name of the NAT pool for which statistic are cleared.

Required Privilege Level clear

List of Sample Output clear services inline nat pool on page 202

Output Fields When you enter this command, the NAT pool statistics are cleared. There is no specific output.

Sample Output

clear services inline nat pool

```
user@host> clear services inline nat pool p1
```


clear services inline nat statistics

Syntax clear services inline nat statistics

Release Information Command introduced in Junos OS Release 11.4.

Description Clear global inline nat statistics

Required Privilege Level clear

List of Sample Output clear services inline nat statistics on page 203

Output Fields When you enter this command, the global inline NAT statistics are cleared. There is no specific output.

Sample Output

clear services inline nat statistics

```
user@host> clear services inline nat statistics
```

show services inline nat pool

Syntax show services inline nat pool
<pool *pool--name*>

Release Information Command introduced in Junos OS Release 11.4.

Description Display information about inline Network Address Translation (NAT) pool.

Options ***pool-name***—Display information about the specified services-inline interface NAT pool.

Required Privilege Level view

List of Sample Output show services inline nat pool on page 204

Output Fields Table 4 on page 204 lists the output fields for the **show services inline nat pool** command. Output fields are listed in the order in which they appear.

Table 4: show services inline nat pool Output Fields

Field Name	Field Description
Interface	Name of an **si** interace hosted on a Trio-based line card.
NAT pool	Name of the pool used for address translations.
Translation type	Translation type specified in the applicable NAT rule for the service set.
Address range	Starting and ending public NAT addresses available for translation.
NATed packets	Number of packets translated for the specified pool.
un-NATed packets	Number of received packets that were not translated.
Errors	Number of packets with translation errors.

Sample Output

show services inline nat pool

```
user@host> show services inline nat pool p1
Interface: si-5/0/0, Service set: ss-inat
  NAT pool: p1, Translation type: BASIC NAT44
    Address range: 20.20.20.0-20.20.20.255
    NATed packets: 0, Un-NATed packets: 0, Errors: 0
```

show services inline nat statistics

Syntax show services inline nat statistics
<interface *interface-name*>

Release Information Command introduced in Junos OS Release 11.4.

Description Display information about inline Network Address Translation (NAT) address translations.

Options ***interface-name***—(Optional) Display information about the specified NAT services-inline interface only. When a specific interface is not specified, statistics for all services-inline interfaces are shown.

Required Privilege Level view

List of Sample Output show services inline nat statistics on page 205

Output Fields Table 5 on page 205 lists the output fields for the **show services inline nat statistics** command. Output fields are listed in the order in which they appear.

Table 5: show services inline nat statistics Output Fields

Field Name	Field Description	Level of Output
Service PIC	Name of an **si** interace hosted on a Trio-based line card.	All levels
Slow path packets received	Number of ICMP exception packets received for NAT translation.	All levels
Slow path packets dropped	Number of received ICMP exception packets that were dropped.	All levels

Sample Output

show services inline nat statistics

```
user@host> show services inline nat statistics
Service PIC Name                                :si-5/0/0

     Slow path packets received                    :0
     Slow path packets dropped                     :0
```

show services nat ipv6-multicast-interfaces

Syntax show services nat ipv6-multicast-interfaces

Release Information Command introduced in Junos OS Release 8.5.

Description Displays a list of interfaces enabled for IPv6 mutlicast.

Required Privilege Level view

List of Sample Output show services nat ipv6-multicast-interfaces on page 206

Output Fields Table 6 on page 206 lists the output fields for the **show services nat ipv6-multicast-interfaces** command. Output fields are listed in the approximate order in which they appear.

Table 6: show services nat ipv6-multicast-interfaces Output Fields

Field Name	Field Description	Level of Output
Interface	Name of a service interface.	All levels
Admin State	Configured IPv6 multicast capability of an interface ,	All levels
Operational State	Operation IPv6 multicast status of an interface.	All levels

Sample Output

show services nat ipv6-multicast-interfaces

```
user@host> show services nat ipv6-multicast-interfaces
Interface          Admin          Operational
                   State             State
ge-5/1/9           Enabled        Enabled
ge-5/1/8           Enabled        Enabled
ge-5/1/7           Enabled        Enabled
ge-5/1/6           Enabled        Enabled
ge-5/1/5           Enabled        Enabled
ge-5/1/4           Enabled        Enabled
ge-5/1/3           Enabled        Enabled
ge-5/1/2           Enabled        Enabled
ge-5/1/1           Enabled        Enabled
ge-5/1/0           Enabled        Enabled
ge-5/0/9           Enabled        Enabled
ge-5/0/8           Enabled        Enabled
ge-5/0/7           Enabled        Enabled
ge-5/0/6           Enabled        Enabled
ge-5/0/5           Enabled        Enabled
ge-5/0/4           Enabled        Enabled
ge-5/0/3           Enabled        Enabled
ge-5/0/2           Enabled        Enabled
ge-5/0/1           Enabled        Enabled
ge-5/0/0           Enabled        Enabled
ge-1/3/9           Enabled        Enabled
ge-1/3/8           Enabled        Enabled
ge-1/3/7           Enabled        Enabled
```



```
ge-1/3/6          Enabled       Enabled
ge-1/3/5          Enabled       Enabled
ge-1/3/4          Enabled       Enabled
ge-1/3/3          Enabled       Enabled
ge-1/3/2          Enabled       Enabled
ge-1/3/1          Enabled       Enabled
ge-1/3/0          Enabled       Enabled
ge-1/2/9          Enabled       Enabled
ge-1/2/8          Enabled       Enabled
ge-1/2/7          Enabled       Enabled
ge-1/2/6          Enabled       Enabled
ge-1/2/5          Enabled       Enabled
ge-1/2/4          Enabled       Enabled
ge-1/2/3          Enabled       Enabled
ge-1/2/2          Enabled       Enabled
ge-1/2/1          Enabled       Enabled
ge-1/2/0          Enabled       Enabled
ge-1/1/9          Enabled       Enabled
ge-1/1/8          Enabled       Enabled
ge-1/1/7          Enabled       Enabled
ge-1/1/6          Enabled       Enabled
ge-1/1/5          Enabled       Enabled
ge-1/1/4          Enabled       Enabled
ge-1/1/3          Enabled       Enabled
ge-1/1/2          Enabled       Enabled
ge-1/1/1          Enabled       Enabled
ge-1/1/0          Enabled       Enabled
ge-1/0/9          Enabled       Enabled
ge-1/0/8          Enabled       Enabled
ge-1/0/7          Enabled       Enabled
ge-1/0/6          Enabled       Enabled
ge-1/0/5          Enabled       Enabled
ge-1/0/4          Enabled       Enabled
ge-1/0/3          Enabled       Enabled
ge-1/0/2          Enabled       Enabled
ge-1/0/1          Enabled       Enabled
ge-1/0/0          Enabled       Enabled
xe-0/3/0          Enabled       Enabled
xe-0/2/0          Enabled       Enabled
xe-0/1/0          Enabled       Enabled
xe-0/0/0          Enabled       Enabled
```

show services nat pool

Syntax
show services nat pool
<brief | detail>
<*pool-name*>
pgcp <ports-per-session | remotely-controlled>

Release Information
Command introduced before Junos OS Release 7.4.
pgcp option added in Junos OS Release 8.5.

Description
Display information about Network Address Translation (NAT) pools.

Options
none—Display standard information about all NAT pools.

brief | detail—(Optional) Display the specified level of output.

pool-name—(Optional) Display information about the specified NAT pool.

pgcp—(Optional) Display information about a NAT pool that is exclusive to the BGF.

ports-per-session—(Optional) Display the number of ports allocated per session from the NAT pool.

remotely-controlled—(Optional) Display if the NAT pool is explicitly specified by the gateway controller.

Required Privilege Level
view

List of Sample Output
show services nat pool brief on page 209
show services nat pool detail on page 209
show services nat pool for Secured Port Block Allocation on page 210
show services nat pool for Deterministic Port Block Allocation on page 210
show services nat pool detail for Port Block Allocation on page 210

Output Fields
Table 7 on page 208 lists the output fields for the **show services nat pool** command. Output fields are listed in the approximate order in which they appear.

Table 7: show services nat pool Output Fields

Field Name	Field Description	Level of Output
Interface	Name of an adaptive services interface.	All levels
Service set	Name of a service set. Individual empty service sets are not displayed, but if none of the service sets has any flows, a flow table header is printed for each service set.	All levels
NAT pool	Name of the Network Address Translation pool.	All levels

Table 7: show services nat pool Output Fields *(continued)*

Field Name	Field Description	Level of Output
Type or **Translation type**	Address translation type: **basic-nat-pt**, **basic-nat44**, **basic-nat66**, **detministic-napt44**, **dnat-44**, **dynamic-nat44**, **napt44**, **napt-66**, **napt-pt**, **stateful-nat64**, **twice-basic-nat-44**, **twice-dynamic-nat-44**, **twice-dynamic-napt-44**.	All levels
Address or **Address range**	IPv4 address range of the pool.	All levels
Port or **Port range**	Port range of the pool. Applicable only for dynamic NAT pools. Not displayed for static NAT pools.	All levels
Ports used' or **Ports in use**	Number of ports allocated in this pool with this name. Applicable only for dynamic NAT pools. Not displayed for static NAT pools.	All levels
Port block type	Type of port block allocation: secured or deterministic	All levels
Out of port errors	Number of port allocation errors. Applicable only for dynamic NAT pools. Not displayed for static NAT pools.	**detail**
Max ports used	Maximum number of ports used. Applicable only for dynamic NAT pools. Not displayed for static NAT pools.	**detail**
Addresses in use	Number of addresses in use for dynamic source address NAT pools.	**detail**

Sample Output

show services nat pool brief

```
user@host> show services nat pool brief

Interface: ms-1/0/0, Service set: s1
NAT pool          Type    Address                         Port        Ports used
dest-pool         DNAT-44 10.10.10.2-10.10.10.2
napt-pool         NAPT-44 50.50.50.1-50.50.50.254         1024-63487  0
source-dynamic-pool DYNAMIC NAT44 40.40.40.1-40.40.40.254
source-static-pool BASIC NAT44 30.30.30.1-30.30.30.254
```

show services nat pool detail

```
user@host> show services nat pool detail

Interface: ms-1/0/0, Service set: s1
  NAT pool: dest-pool, Translation type: DNAT-44
    Address range: 10.10.10.2-10.10.10.2
  NAT pool: napt-pool, Translation type: NAPT-44
    Address range: 50.50.50.1-50.50.50.254
    Port range: 1024-63487, Ports in use: 0, Out of port errors: 0, Max ports
used: 0
  NAT pool: source-dynamic-pool, Translation type: DYNAMIC NAT44
    Address range: 40.40.40.1-40.40.40.254
    Out of address errors: 0, Addresses in use: 0
  NAT pool: source-static-pool, Translation type: BASIC NAT44
    Address range: 30.30.30.1-30.30.30.254
```

show services nat pool for Secured Port Block Allocation

```
user@host> show services nat pool
Interface: sp-2/0/0, Service set: in
NAT pool          Type    Address                          Port        Ports used
mypool            dynamic 3.3.3.3-3.3.3.10                 512-65535   0
                          3.3.3.15-3.3.3.20
                          3.3.3.25-3.3.3.30
                          3.3.3.95-3.3.3.200
Port block size: 64, Max port blocks per address: 1, Active block timeout: 86400,
 Effective port range: 1024-65471,
Effective number of port blocks: 126882, Effective number of ports: 8120448, Port
 block efficiency: nan

Interface: sp-2/1/0, Service set: in1
NAT pool          Type    Address                          Port        Ports used
mypool1           dynamic 9.9.9.1-9.9.9.254                512-65535   0
Port block size: 64, Max port blocks per address: 1, Active block timeout: 86400,
 Effective port range: 1024-65471,
Effective number of port blocks: 255778, Effective number of ports: 16369792,
Port block efficiency: nan
```

show services nat pool for Deterministic Port Block Allocation

```
user@host> show services nat pool

Interface: sp-2/0/0, Service set: ss2
NAT pool          Type    Address                          Port        Ports Used
pba               dynamic 33.33.33.1-33.33.33.128          512-65535   6604
Port block type: Deterministic port block, Port block size: 200
```

show services nat pool detail for Port Block Allocation

```
user@host> show services nat pool detail

Interface: sp-0/0/0, Service set: in
  NAT pool: x, Translation type: dynamic
    Address range: 1.1.1.1-1.1.1.1
    Address range: 4.4.4.4-4.4.4.40
   Port range: 512-65535, Ports in use: 0, Out of port errors: 0, Max ports used:
 0
    Max number of port blocks used: 0, Current number of port blocks in use: 0,
Port block allocation errors: 0
```

show services nat mapping

Syntax
show services nat mapping
<brief | detail | summary>
<*pool-name*>

Release Information
Command introduced in Junos OS Release 10.1.
summary option introduced in Junos OS Release 11.1.

Description
Display information about Network Address Translation (NAT) address and port mappings.

Options
none—Display standard information about all NAT pools.

brief | detail | summary—(Optional) Display the specified level of output.

pool-name—(Optional) Display information about the specified NAT pool.

Required Privilege Level
view

List of Sample Output
show services nat mapping brief on page 212
show services nat mapping detail on page 212
show services nat mapping pool-name on page 212
show services nat mapping summary on page 212

Output Fields
Table 8 on page 211 lists the output fields for the **show services nat mapping** command. Output fields are listed in the approximate order in which they appear.

Table 8: show services nat mapping Output Fields

Field Name	Field Description	Level of Output
Interface	Name of a service interface.	All levels
Service set	Name of a service set. Individual empty service sets are not displayed, but if none of the service sets has any flows, a flow table header is printed for each service set.	All levels
NAT pool	Name of the NAT pool.	All levels
Address Mapping	Mapping performed by NAT to conceal the network address.	All levels
No. of Port Mappings	Number of port mappings.	All levels
Port mapping	Port mapping performed by NAT.	**detail**
Flow Count	Number of flows.	**detail**
Total number of address mappings:	Total number of address mappings for all interfaces and service sets.	**summary**

Table 8: show services nat mapping Output Fields *(continued)*

Field Name	Field Description	Level of Output
Total number of endpoint independent port mappings:	Total number of port mappings for interfaces and services sets.	**summary**
Total number of endpoint independent filters:	Total number of independent filters that filter out only packets that are not destined to the internal address and port regardless of the external IP address and port source.	**summary**

Sample Output

show services nat mapping brief

```
user@host> show services nat mapping brief
Interface: sp-2/3/0, Service set: s1

    NAT pool: p1
      Address Mapping: 2.1.20.10 ---> 34.34.34.34
        No. of port mappings: 1
```

show services nat mapping detail

```
user@host> show services nat mapping detail
Interface: sp-2/3/0, Service set: s1

    NAT pool: p1
      Address Mapping: 2.1.20.10 ---> 34.34.34.34, No. of port mappings: 1
        Port mapping: 49604 -->  1024, Flow Count: 2
```

show services nat mapping pool-name

```
user@host> show services nat mapping p1
Interface: sp-2/3/0, Service set: s1

    NAT pool: p1
      Address Mapping: 2.1.20.10 ---> 34.34.34.34
        No. of port mappings: 1
```

show services nat mapping summary

```
user@host> show services nat mapping summary
Total number of address mappings:                       500000
Total number of endpoint independent port mappings:     500000
Total number of endpoint independent filters:           0
```

show services softwire

Syntax **show services softwire**
<count>

Release Information Command introduced in Junos OS Release 10.4.
<count> option added in Junos OS Release 11.2.

Description Display information about softwire services. Information is displayed on both 6rd and DS-Lite services.

Options **count** ***interface-name*** —(Optional) Display the current softwire counts for a service set for both DS-Lite and 6rd.

Required Privilege Level view

List of Sample Output show services softwire on page 213
show services softwire count on page 213

Output Fields Table 9 on page 213 lists the output fields for the **command-name** command. Output fields are listed in the approximate order in which they appear.

Table 9: show-services-softwire Output Fields

Field Name	Field Description	Level of Output
Interface	Interface for which information is displayed.	All levels
Service Set	Service set containing the softwire rules for the interface.	All levels
Softwire	Name of the softwire concentrator.	All levels
Direction	Direction of the flow.	All levels
Flow count	Number of flows.	All levels

Sample Output

show services softwire

```
user@host> show services softwire
Interface: sp-3/0/0, Service set: v6rd-dom1-dom3-service-set
Softwire                                    Direction     Flow count
10.10.10.2      ->     30.30.30.1              I                 13
```

show services softwire count

```
user@host> show services softwire count
Interface   Service set                  DS-Lite        6RD
sp-0/0/0    dslite-svc-set1              2              0
```

show services softwire flows

Syntax

show services softwire flows
(<interface ***interface-name*****> <service-set** ***service-set-name*****>|**
count <interface ***interface-name*****> <service-set** ***service-set-name*****>|**
ds-lite <B4 ***b4-address*****> <AFTR** ***aftr-address*****>|**
v6rd <initiator ***initiator-ip-address*****><concentrator** ***concentrator-ip-address*****>)**

Release Information

Command introduced in Junos OS Release 10.2.

Description

Display statistics information about the softwire flows.

Options

interface ***interface-name***—(Optional) Display statistics information about the specified interface only.

service-set ***service-set-name***—(Optional) Display statistics information about the specified service set only.

count <interface ***interface-name*****> <service-set** ***service-set-name*****>|**—(Optional) Display flow count information only, with optional filtering by interface and service set.

ds-lite <B4 ***b4-address*****> <AFTR** ***aftr-address*****>|**—(Optional) Display DS-Lite flow information, with optional filtering by B4 (softwire initiator) and AFTR (softwire concentrator).

v6rd <initiator ***initiator-ip-address*****><concentrator** ***concentrator-ip-address*****>)**—(Optional) Display v6rd flow information, with optional filtering by the softwire initiator and softwire concentrator.

Required Privilege Level

view

List of Sample Output

show services softwire flows on page 215
show services softwire flows count on page 215
show services softwire flows ds-lite B4 on page 215
show services softwire flows ds-lite AFTR on page 216
services softwire flows ds-lite AFTR and B4 on page 216

Output Fields

Table 10 on page 214 lists the output fields for the **show services softwire flows** command. Output fields are listed in the approximate order in which they appear.

Table 10: show services softwire flows Output Fields

Field Name	Field Description
Interface	Name of the interface.
Service set	Name of the service set.
Flow	Description of flow, including protocol input and output interface addresses.

Table 10: show services softwire flows Output Fields *(continued)*

Field Name	Field Description
State	Flow state. Value is: • **Forward**
Dir	Flow direction. Values are: • **I**—inbound • **O**—outbound
Frm count	Number of frames transferred.
NAT dest	NAT translation of the decapsulated address.
Softwire	For outbound flows, the address of the local softwire initiator (B4 for DS-Lite) is shown first, followed by the address of the softwire concentrator (AFTR for DS-Lite). For inbound flows, the address of the software concentrator is shown first, followed by the address of the softwire initiator.

Sample Output

show services softwire flows

```
user@host> show services softwire flows
Interface: sp-0/0/0, Service set: dslite-svc-set1
Flow                                              State       Dir       Frm count
TCP      200.200.200.2:80    ->     33.33.33.1:1066  Forward  O         2005418
    NAT dest        33.33.33.1:1066    ->       20.20.1.2:1025
    Softwire          1001::1          ->         2001::2
TCP          20.20.1.2:1025  ->  200.200.200.2:80    Forward  I         2007168
    NAT source       20.20.1.2:1025    ->      33.33.33.1:1066
    Softwire          2001::2          ->         1001::1
TCP          20.20.1.2:1025  ->  200.200.200.2:80    Forward  I         2635998
    NAT source       20.20.1.2:1025    ->      33.33.33.1:1065
    Softwire          2001::3          ->         1001::1
DS-LITE         2001::2      ->        1001::1       Forward  I         2008157
TCP      200.200.200.2:80    ->     33.33.33.1:1065  Forward  O         2637909
    NAT dest        33.33.33.1:1065    ->       20.20.1.2:1025
    Softwire          1001::1          ->         2001::3
DS-LITE         2001::3      ->        1001::1       Forward  I         2640499
```

show services softwire flows count

```
user@host> show services softwire flows count
Interface   Service set                                                 Flow count
sp-0/0/0    dslite-svc-set1                                             6
```

show services softwire flows ds-lite B4

```
user@host> show services softwire flows ds-lite B4 2001::2
Interface: sp-0/0/0, Service set: dslite-svc-set1
Flow                                              State       Dir       Frm count
TCP      200.200.200.2:80    ->     33.33.33.1:1066  Forward  O         2884037
    NAT dest        33.33.33.1:1066    ->       20.20.1.2:1025
    Softwire          1001::1          ->         2001::2
TCP          20.20.1.2:1025  ->  200.200.200.2:80    Forward  I         2885884
    NAT source       20.20.1.2:1025    ->      33.33.33.1:1066
```

```
    Softwire           2001::2         ->         1001::1
DS-LITE         2001::2      ->        1001::1       Forward  I         2886821
```

show services softwire flows ds-lite AFTR

```
user@host> show services softwire flows ds-lite AFTR 1001::1
Interface: sp-0/0/0, Service set: dslite-svc-set1
Flow                                                State       Dir       Frm count
TCP      200.200.200.2:80    ->     33.33.33.1:1066  Forward  O         3359356
    NAT dest        33.33.33.1:1066    ->       20.20.1.2:1025
    Softwire           1001::1         ->         2001::2
TCP          20.20.1.2:1025  ->  200.200.200.2:80    Forward  I         3361235
    NAT source       20.20.1.2:1025    ->      33.33.33.1:1066
    Softwire           2001::2         ->         1001::1
TCP          20.20.1.2:1025  ->  200.200.200.2:80    Forward  I         4479810
    NAT source       20.20.1.2:1025    ->      33.33.33.1:1065
    Softwire           2001::3         ->         1001::1
DS-LITE         2001::2      ->        1001::1       Forward  I         3362168
TCP      200.200.200.2:80    ->     33.33.33.1:1065  Forward  O         4481520
    NAT dest        33.33.33.1:1065    ->       20.20.1.2:1025
    Softwire           1001::1         ->         2001::3
DS-LITE         2001::3      ->        1001::1       Forward  I         4484094
```

services softwire flows ds-lite AFTR and B4

```
user@host> show services softwire flows ds-lite AFTR 1001::1 B4 2001::2
Interface: sp-0/0/0, Service set: dslite-svc-set1
Flow                                                State       Dir       Frm count
TCP      200.200.200.2:80    ->     33.33.33.1:1066  Forward  O         3931026
    NAT dest        33.33.33.1:1066    ->       20.20.1.2:1025
    Softwire           1001::1         ->         2001::2
TCP          20.20.1.2:1025  ->  200.200.200.2:80    Forward  I         3932792
    NAT source       20.20.1.2:1025    ->      33.33.33.1:1066
    Softwire           2001::2         ->         1001::1
DS-LITE         2001::2      ->        1001::1       Forward  I         3933782
```

show services softwire statistics

Syntax
<ds-lite>
<inferface *interface-name*>
<v6rd>

Release Information Command introduced in JUNOS Release 10.4.

Description Display information about softwire services.

Options
ds-lite—(Optional) Display only DS-Lite.

interface ***interface-name*** —(Optional) Name of the interface servicing the softwire. When you omit this option, data for all interfaces are shown.

v6rd—(Optional) Display only 6rd statistics.

Required Privilege Level view

List of Sample Output
show services softwire on page 219
show services softwire statistics on page 219

Output Fields Table 11 on page 217 lists the output fields for the **command-name** command. Output fields are listed in the approximate order in which they appear.

Table 11: command-name Output Fields

Field Name	Field Description	Level of Output
Interface	Interface for which information is displayed.	All levels
Service Set	Service set containing the softwire rules for the interface.	All levels
Softwire	Name of the softwire concentrator.	All levels
Direction	Direction of the flow.	All levels
Flow count	Number of flows.	All levels
Softwires Created	Number of softwires created.	**statistics**
Softwires Deleted	Number of softwires deleted.	**statistics**
Flows Created	Number of flows created.	**statistics**
Flows Deleted	Number of flows deleted.	**statistics**
Slow Path	Number of packets processed as initial packets in a softwire session. These packets require a rule lookup and setting up of flows; this processing of an initial packet in a flow is called “the slow path.”	**statistics**

Table 11: command-name Output Fields *(continued)*

Field Name	Field Description	Level of Output
Fast Path	Number of packets processed that are not "slow path."	**statistics**
Fast Path Encapsulated	Number of packets encapsulated in the fast path.	**statistics**
Rule Match Failed	Number of packets that did not match any softwire rule.	**DS-Lite** and **6rd**
Rule Match Succeeded	Number of packets that matched a softwire rule.	**statistics**
IPv6 Packets Fragmented	Number of packets fragmented by the services PIC.	**DS-Lite**
Softwire Creation Failed	Number of softwire creation failures.	**DS-Lite** and **6rd**
Flow Creation Failed	Number of flow creation failures.	**statistics**
Flow Creation Failed - Retry	Number of flows creations retried after failure.	**statistics**
Slow Path Failed	Number of failures detected in the slow path.	**statistics**
Slow Path Failed - Retry	Number of times processing of a packet was reprocessed in the slow path.	**statistics**
Packet not IPv4 in IPv6	Number of IPv4 packets not encapsulated in IPv6.	**statistics** for **ds-lite** only
Slow Path Failed - IPv6 Next Header Offset	Number of IPv6 header errors detected in slow path processing.	**statistics** for **ds-lite** only
Decapsulated Packet not IPv4	Number of packets without IPv4 inner header.	**statistics** for **ds-lite** only
Fast Path Failed - IPv6 Next Header Offset	Number of IPv6 header errors detected in fast path processing.	**statistics** for **ds-lite** only
No Softwire ID	Number of times a softwire ID was not found.	**statistics**
No Flow Extension	Number of times flow extensions were not found.	**statistics**
Packet not IPv6 in IPv4	Number of IPv6 packets not encapsulated in IPv4.	**statistics** for **v6rd** only

Table 11: command-name Output Fields *(continued)*

Field Name	Field Description	Level of Output
Decapsulated Packet not IPv6	Number of packets without IPv6 inner header.	**statistics** for **v6rd** only
Encapsulation Failed - No space for Outer Header	Failed to encapsulate IPv6 packets in IPv4 due to low memory.	**statistics** for **v6rd** only

Sample Output

show services softwire

```
user@host> show services softwire
Interface: sp-3/0/0, Service set: v6rd-dom1-dom3-service-set
Softwire                                        Direction     Flow count
10.10.10.2      ->     30.30.30.1                  I                 13
```

show services softwire statistics

```
user@host> show services softwire statistics
DS-Lite Statistics:


  Service PIC Name:                             :sp-0/0/0

  Statistics
  ----------

    Softwires Created                           :2
    Softwires Deleted                           :0
    Softwires Flows Created                     :2
    Softwires Flows Deleted                     :0
    Slow Path Packets Processed                 :2
    Fast Path Packets Processed                 :1043786
    Fast Path Packets Encapsulated              :1450803
    Rule Match Failed                           :0
    Rule Match Succeeded                        :2
    IPv6 Packets Fragmented                     :0

  Transient Errors
  ----------------

    Flow Creation Failed - Retry                :0
    Slow Path Failed - Retry                    :0

  Errors
  ------

    Softwire Creation Failed                    :0
    Flow Creation Failed                        :0
    Slow Path Failed                            :0
    Packet not IPv4-in-IPv6                     :0
    IPv6 Fragmentation Error                    :0
    Slow Path Failed - IPv6 Next Header Offset  :0
    Decapsulated Packet not IPv4                :0
    Fast Path Failed - IPv6 Next Header Offset  :0
    No Softwire ID                              :0
    No Flow Extension                           :0
    Flow Limit Exceeded                         :0
```

```
6rd Statistics:


 Service PIC Name                                :sp-0/0/0

  Statistics
  ----------

    Softwires Created                            :0
    Softwires Deleted                            :0
    Softwires Flows Created                      :0
    Softwires Flows Deleted                      :0
    Slow Path Packets Processed                  :0
    Fast Path Packets Processed                  :0
    Fast Path Packets Encapsulated               :0
    Rule Match Failed                            :0
    Rule Match Succeeded                         :0

  Transient Errors
  ----------------

    Flow Creation Failed - Retry                 :0
    Slow Path Failed - Retry                     :0

  Errors
  ------

    Softwire Creation Failed                     :0
    Flow Creation Failed                         :0
    Slow Path Failed                             :0
    Packet not IPv6-in-IPv4                      :0
    Slow Path Failed - IPv6 Next Header Offset   :0
    Decapsulated Packet not IPv6                 :0
    Encapsulation Failed - No packet memory      :0
    No Softwire ID                               :0
    No Flow Extension                            :0
    ICMPv4 Dropped Packets                       :0
```

show services stateful-firewall conversations

Syntax show services stateful-firewall conversations
<brief | extensive | terse>
<application-protocol *protocol*>
<destination-port *destination-port*>
<destination-prefix *destination-prefix*>
<interface *interface-name*>
<limit *number*>
<pgcp>
<protocol *protocol*>
<service-set *service-set*>
<source-port *source-port*>
<source-prefix *source-prefix*>

Release Information Command introduced before Junos OS Release 7.4.
pgcp option introduced in Junos OS Release 8.4.

Description Display information about stateful firewall conversations.

Options **none**—Display standard information about all stateful firewall conversations.

brief | extensive | terse—(Optional) Display the specified level of output.

application-protocol *protocol*—(Optional) Display information about one of the following application protocols:

- **bootp**—Bootstrap protocol
- **dce-rpc**—Distributed Computing Environment-Remote Procedure Call protocols
- **dce-rpc-portmap**—Distributed Computing Environment-Remote Procedure Call protocols portmap service
- **dns**—Domain Name System protocol
- **exec**—Exec
- **ftp**—File Transfer Protocol
- **h323**—H.323 standards
- **icmp**—Internet Control Message Protocol
- **iiop**—Internet Inter-ORB Protocol
- **login**—Login
- **netbios**—NetBIOS
- **netshow**—NetShow
- **realaudio**—RealAudio
- **rpc**—Remote Procedure Call protocol
- **rpc-portmap**—Remote Procedure Call protocol portmap service
- **rtsp**—Real-Time Streaming Protocol

- **shell**—Shell
- **sip**—Session Initiation Protocol
- **snmp**—Simple Network Management Protocol
- **sqlnet**—SQLNet
- **tftp**—Trivial File Transfer Protocol
- **traceroute**—Traceroute
- **winframe**—WinFrame

destination-port ***destination-port***—(Optional) Display information for a particular destination port. The range of values is 0 to 65535.

destination-prefix ***destination-prefix***—(Optional) Display information for a particular destination prefix.

interface ***interface-name***—(Optional) Display information about a particular interface. On M Series and T Series routers, the ***interface-name*** can be **sp-*fpc/pic/port*** or **rsp*number***. On J Series routers, the ***interface-name*** is **sp-*pim*/0/*port***.

limit ***number***—(Optional) Maximum number of entries to display.

pgcp —(Optional) Display information about stateful firewall conversations for Packet Gateway Control Protocol (PGCP) flows.

protocol ***protocol***—(Optional) Display information about one of the following IP types:

- ***number***—Numeric protocol value from 0 to 255
- **ah**—IPsec Authentication Header protocol
- **egp**—An exterior gateway protocol
- **esp**—IPsec Encapsulating Security Payload protocol
- **gre**—A generic routing encapsulation protocol
- **icmp**—Internet Control Message Protocol
- **igmp**—Internet Group Management Protocol
- **ipip**—IP-within-IP Encapsulation Protocol
- **ospf**—Open Shortest Path First protocol
- **pim**—Protocol Independent Multicast protocol
- **rsvp**—Resource Reservation Protocol
- **sctp**—Stream Control Protocol
- **tcp**—Transmission Control Protocol
- **udp**—User Datagram Protocol

service-set ***service-set***—(Optional) Display information for the specific service set.

source-port *source-port*—(Optional) Display information for a particular source port. The range of values is 0 to 65535.

source-prefix *source-prefix*—(Optional) Display information for a particular source prefix.

Required Privilege Level view

List of Sample Output
show services stateful-firewall conversations on page 224
show services stateful-firewall conversations destination-port on page 224

Output Fields Table 12 on page 223 lists the output fields for the **show services stateful-firewall conversations** command. Output fields are listed in the approximate order in which they appear.

Table 12: show services stateful-firewall conversations Output Fields

Field Name	Field Description
Interface	Name of an adaptive services interface.
Service set	Name of a service set. Individual empty service sets are not displayed, but if no service set has any flows, a flow table header is printed for each service set.
Conversation	Information about a group of related flows. • **ALG Protocol**—Application-level gateway protocol. • **Number of initiators**—Number of flows that initiated a session. • **Number of responders**—Number of flows that responded in a session.
Flow or Flow Prot	Protocol used for this flow.
Source	Source prefix of the flow, in the format ***source-prefix-port***.
Destination	Destination prefix of the flow.
State	Status of the flow: • **Drop**—Drop all packets in the flow without response. • **Forward**—Forward the packet in the flow without looking at it. • **Reject**—Drop all packets in the flow with response. • **Watch**—Inspect packets in the flow.
Dir	Direction of the flow: input (**I**) or output (**O**).
Source NAT	Original and translated source IPv4 or IPv6 addresses are displayed if Network Address Translation (NAT) is configured on this particular flow or conversation.
Frm Count	Number of frames in the flow.
Destin NAT	Original and translated destination IPv4 or IPv6 addresses are displayed if NAT is configured on this particular flow or conversation.

Table 12: show services stateful-firewall conversations Output Fields *(continued)*

Field Name	Field Description
Byte count	Number of bytes forwarded in the flow.
TCP established	Whether a TCP connection was established: **Yes** or **No**.
TCP window size	Negotiated TCP connection window size, in bytes.
TCP acknowledge	TCP acknowledgment sequence number.
TCP tickle	Whether TCP inquiry mode is on (**enabled** or **disabled**) and the time remaining to send the next inquiry, in seconds.
Master flow	Flow that initiated the conversation.
TImeout	Lifetime of the flow, in seconds.

Sample Output

show services stateful-firewall conversations

```
user@host> show services stateful-firewall conversations
Interface: sp-1/3/0, Service set: green
Conversation: ALG Protocol: any, Number of initiators: 1,
Number of responders: 1

Flow
Prot     Source                Dest               State      Dir   Frm count
TCP      10.58.255.50:33005->   10.58.255.178:23   Forward    I      13
  Source NAT     10.58.255.50:33005->    10.59.16.100:4000
  Destin NAT     10.58.255.178:23  ->         0.0.0.0:4000
Byte count:          918
TCP established, TCP window size: 65535, TCP acknowledge: 2502627025
TCP tickle enabled, 0 seconds,
Master flow, Timeout: 30 seconds
TCP     10.58.255.178:23   ->    10.59.16.100:4000 Forward    0        8
```

show services stateful-firewall conversations destination-port

```
user@host> show services stateful-firewall conversations destination-port 21
Interface: sp-0/3/0, Service set: svc_set_trust

Interface: sp-0/3/0, Service set: svc_set_untrust
Conversation: ALG protocol: ftp
  Number of initiators: 1, Number of responders: 1
Flow                                                 State    Dir       Frm count
TCP         10.50.10.2:2143  ->     10.50.20.2:21    Watch    O              0
TCP         10.50.20.2:21    ->     10.50.10.2:2143  Watch    I              0
TCP         10.50.20.2:21    ->     10.50.10.2:2143  Watch    I              0
```

show services stateful-firewall flows

Syntax

show services stateful-firewall flows
<brief | extensive | summary | terse>
<application-protocol *protocol*>
<count>
<destination-port *destination-port*>
<destination-prefix *destination-prefix*>
<interface *interface-name*>
<limit *number*>
<protocol *protocol*>
<service-set *service-set*>
<source-port *source-port*>
<source-prefix *source-prefix*>

Release Information

Command introduced before Junos OS Release 7.4.
pgcp option introduced in Junos OS Release 8.4.
application-protocol option introduced in Junos OS Release 10.4.

Description

Display stateful firewall flow table entries. When the interface is used for softwire processing, the type of softwire concentrator (**DS-LITE** or **6rd**) is shown, and frame counts are provided.

Options

none—Display standard information about all stateful firewall flows.

brief | extensive | summary | terse—(Optional) Display the specified level of output.

application-protocol *application-protocol*—(Optional) Display information about one of the following application-level gateway (ALG) protocol types:

- **bootp**—Bootstrap protocol
- **dce-rpc**—Distributed Computing Environment (DCE) remote procedure call (RPC) protocol

NOTE: Use this option to select Microsoft Remote Procedure Call (MSRPC).

- **dce-rpc-portmap**—Distributed Computing Environment (DCE) remote procedure call (RPC) portmap protocol
- **dns**—Domain Name Service protocol
- **exec**—Remote execution protocol
- **ftp**—File Transfer Protocol
- **h323**—H.323 protocol
- **icmp**—Internet Control Message Protocol
- **iiop**—Internet Inter-ORB Protocol

- **ip**—Internet protocol
- **netbios**—NetBIOS protocol
- **netshow**—Netshow protocol
- **pptp** —Point-to-Point Tunneling Protocol
- **realaudio**—RealAudio protocol
- **rpc**—Remote Procedure Call protocol

NOTE: Use this option to select Sun Microsystems Remote Procedure Call protocol (SunRPC).

- **rpc-portmap**—Remote Procedure Call portmap protocol
- **rtsp**—Real-Time Streaming Protocol
- **sip**—Session Initiation Protocol
- **snmp**—Simple Network Management Protocol
- **talk**—Talk protocol
- **tftp**—Trivial File Transfer Protocol
- **traceroute**—Traceroute
- **winframe**—WinFrame

count—(Optional) Display a count of the matching entries.

destination-port ***destination-port***—(Optional) Display information for a particular destination port. The range of values is from 0 to 65535.

destination-prefix ***destination-prefix***—(Optional) Display information for a particular destination prefix.

interface ***interface-name***—(Optional) Display information about a particular interface. On M Series and T Series routers, ***interface-name*** can be **ms-*fpc/pic/port*** or **rsp*number***. On J Series routers, ***interface-name*** is **ms-*pim*/0/*port***.

limit ***number***—(Optional) Maximum number of entries to display.

protocol ***protocol***—(Optional) Display information about one of the following IP types:

- ***number***—Numeric protocol value from 0 to 255
- **ah**—IPsec Authentication Header protocol
- **egp**—An exterior gateway protocol
- **esp**—IPsec Encapsulating Security Payload protocol
- **gre**—A generic routing encapsulation protocol

- **icmp**—Internet Control Message Protocol
- **igmp**—Internet Group Management Protocol
- **ipip**—IP-within-IP Encapsulation Protocol
- **ospf**—Open Shortest Path First protocol
- **pim**—Protocol Independent Multicast protocol
- **rsvp**—Resource Reservation Protocol
- **sctp**—Stream Control Protocol
- **tcp**—Transmission Control Protocol
- **udp**—User Datagram Protocol

service-set *service-set*—(Optional) Display information for a particular service set.

source-port *source-port*—(Optional) Display information for a particular source port. The range of values is from 0 to 65535.

source-prefix *source-prefix*—(Optional) Display information for a particular source prefix.

Required Privilege Level view

Related Documentation
- clear services stateful-firewall flows

List of Sample Output
show services stateful-firewall flows on page 228
show services stateful-firewall flows (For Softwire Flows) on page 228
show services stateful-firewall flows brief on page 229
show services stateful-firewall flows extensive on page 229
show services stateful-firewall flows count on page 229
show services stateful-firewall flows destination port on page 229
show services stateful-firewall flows source port on page 229
show services stateful-firewall flows (Twice NAT) on page 229

Output Fields Table 13 on page 227 lists the output fields for the **show services stateful-firewall flows** command. Output fields are listed in the approximate order in which they appear.

Table 13: show services stateful-firewall flows Output Fields

Field Name	Field Description
Interface	Name of the interface.
Service set	Name of a service set. Individual empty service sets are not displayed. If no service set has any flows, a flow table header is displayed for each service set.
Flow Count	Number of flows in a session.
Flow or Flow Prot	Protocol used for this flow.

Table 13: show services stateful-firewall flows Output Fields *(continued)*

Field Name	Field Description
Source	Source prefix of the flow in the format ***source-prefix:port***. For ICMP flows, port information is not displayed.
Dest	Destination prefix of the flow. For ICMP flows, port information is not displayed.
State	Status of the flow: • **Drop**—Drop all packets in the flow without response. • **Forward**—Forward the packet in the flow without looking at it. • **Reject**—Drop all packets in the flow with response. • **Watch**—Inspect packets in the flow.
Dir	Direction of the flow: input **(I)** or output **(O)**.
Frm count	Number of frames in the flow.

Sample Output

show services stateful-firewall flows

```
user@host> show services stateful-firewall flows
Interface: ms-1/3/0, Service set: green

Flow
Prot     Source                Dest                State      Dir     Frm count
TCP     10.58.255.178:23   ->    10.59.16.100:4000 Forward    O
TCP      10.58.255.50:33005->   10.58.255.178:23   Forward    I             1
  Source NAT    10.58.255.50:33005->    10.59.16.100:4000
  Destin NAT    10.58.255.178:23   ->         0.0.0.0:4000
```

show services stateful-firewall flows (For Softwire Flows)

When a service set includes softwire processing, the following output format is used for the softwire flows:

```
user@host> show services stateful-firewall flows
Interface: sp-0/1/0, Service set: dslite-svc-set2
Flow                                                 State    Dir       Frm count
TCP      200.200.200.2:80    ->     44.44.44.1:1025  Forward  O          219942
    NAT dest        44.44.44.1:1025    ->       20.20.1.4:1025
    Softwire          2001::2          ->         1001::1
TCP          20.20.1.2:1025  ->  200.200.200.2:80    Forward  I          110244
    NAT source       20.20.1.2:1025    ->      44.44.44.1:1024
    Softwire          2001::2          ->         1001::1
TCP      200.200.200.2:80    ->     44.44.44.1:1024  Forward  O          219140
    NAT dest        44.44.44.1:1024    ->       20.20.1.2:1025
    Softwire          2001::2          ->         1001::1
DS-LITE         2001::2      ->        1001::1       Forward  I          988729
TCP      200.200.200.2:80    ->     44.44.44.1:1026  Forward  O          218906
    NAT dest        44.44.44.1:1026    ->       20.20.1.3:1025
    Softwire          2001::2          ->         1001::1
TCP          20.20.1.3:1025  ->  200.200.200.2:80    Forward  I          110303
    NAT source       20.20.1.3:1025    ->      44.44.44.1:1026
    Softwire          2001::2          ->         1001::1
TCP          20.20.1.4:1025  ->  200.200.200.2:80    Forward  I          110944
```

```
    NAT source       20.20.1.4:1025    ->      44.44.44.1:1025
    Softwire           2001::2         ->         1001::1
```

show services stateful-firewall flows brief

The output for the **show services stateful-firewall flows brief** command is identical to that for the **show services stateful-firewall flows** command. For sample output, see show services stateful-firewall flows.

show services stateful-firewall flows extensive

```
user@host> show services stateful-firewall flows extensive
Interface: ms-0/3/0, Service set: ss_nat
Flow                                                  State     Dir       Frm
count
TCP           16.1.0.1:2330  ->      16.49.0.1:21        Forward    I
   8
    NAT source        16.1.0.1:2330    ->       16.41.0.1:2330
    NAT dest         16.49.0.1:21      ->       16.99.0.1:21
  Byte count: 455, TCP established, TCP window size: 57344
  TCP acknowledge: 3251737524, TCP tickle enabled, tcp_tickle: 0
  Flow role: Master, Timeout: 720
TCP          16.99.0.1:21    ->      16.41.0.1:2330      Forward    O
   5
    NAT source       16.99.0.1:21      ->       16.49.0.1:21
    NAT dest         16.41.0.1:2330    ->        16.1.0.1:2330
  Byte count: 480, TCP established, TCP window size: 57344
  TCP acknowledge: 463128048, TCP tickle enabled, tcp_tickle: 0
  Flow role: Responder, Timeout: 720
```

show services stateful-firewall flows count

```
user@host> show services stateful-firewall flows count
Interface             Service set                                  Flow Count

ms-1/3/0              green                                                 2
```

show services stateful-firewall flows destination port

```
user@router> show services stateful-firewall flows destination-port 21
Interface: ms-0/3/0, Service set: svc_set_trust
Flow
                                                 State    Dir       Frm count
Interface: ms-0/3/0, Service set: svc_set_untrust
Flow                                            State    Dir       Frm count
TCP         10.50.10.2:2143  ->     10.50.20.2:21    Watch    O               0
```

show services stateful-firewall flows source port

```
user@router> show services stateful-firewall flows source-port 2143
Interface: ms-0/3/0, Service set: svc_set_trust
Flow
                                              State    Dir       Frm count
Interface: ms-0/3/0, Service set: svc_set_untrust
Flow                                            State    Dir       Frm count
TCP         10.50.10.2:2143  ->     10.50.20.2:21    Watch    O               0
```

show services stateful-firewall flows (Twice NAT)

```
user@router> show services stateful-firewall flows
Flow                                          State    Dir       Frm count
UDP          40.0.0.8:23439 ->     80.0.0.1:16485   Watch    I             20
    NAT source        40.0.0.8:23439   ->     172.16.1.10:1028
    NAT dest          80.0.0,1:16485   ->     192.16.1.10:22415
UDP      192.16.1.10:22415  ->  172.16.1.10:1028    Watch    O             20
    NAT source     192.16.1.10:22415   ->        80.0.0.1:16485
    NAT dest       172.16.1.10:1028    ->        40.0.0.8:23439
```

show services stateful-firewall statistics

Syntax

show services stateful-firewall statistics
<application-protocol *protocol*>
<brief | detail | extensive | summary>
<interface *interface-name*>
<service-set *service-set*>

Release Information

Command introduced before Junos OS Release 7.4.

Description

Display stateful firewall statistics.

Options

none—Display standard information about all stateful firewall statistics.

brief | detail | extensive | summary—(Optional) Display the specified level of output.

interface *interface-name*—(Optional) Display information about a particular interface. On M Series and T Series routers, the ***interface-name*** can be **ms-*fpc/pic/port*** or **rsp*number***. On J Series routers, the ***interface-name*** is **ms-*pim*/0/*port***.

service-set *service-set*—(Optional) Display information about a particular service set.

Required Privilege Level

view

Related Documentation

- clear services stateful-firewall statistics

List of Sample Output

show services stateful-firewall statistics extensive on page 233

Output Fields

Table 14 on page 230 lists the output fields for the **show services stateful-firewall statistics** command. Output fields are listed in the approximate order in which they appear.

Table 14: show services stateful-firewall statistics Output Fields

Field Name	Field Description
Interface	Name of an adaptive services interface.
Service set	Name of a service set.
New flows	Rule match counters for new flows: • **Accept**—New flows accepted. • **Discard**—New flows discarded. • **Reject**—New flows rejected.
Existing flows	Rule match counters for existing flows: • **Accept**—Match existing forward or watch flow. • **Discard**—Match existing discard flow. • **Reject**—Match existing reject flow.

Table 14: show services stateful-firewall statistics Output Fields *(continued)*

Field Name	Field Description
Drops	Drop counters: • **TCP SYN defense**—Packets dropped by SYN defender. • **NAT ports exhausted**—Hide mode. The router has no available Network Address Translation (NAT) ports for a given address or pool.
Errors	Total errors, categorized by protocol: • **IP**—Total IP version 4 errors. • **TCP**—Total Transmission Control Protocol (TCP) errors. • **UDP**—Total User Datagram Protocol (UDP) errors. • **ICMP**—Total Internet Control Message Protocol (ICMP) errors. • **Non-IP**—Total non-IPv4 errors.
IP Errors	IPv4 errors: • **IP packet length inconsistencies**—IP packet length does not match the Layer 2 reported length. • **Minimum IP header length check failures**—Minimum IP header length is 20 bytes. The received packet contains less than 20 bytes. • **Reassembled packet exceeds maximum IP length**—After fragment reassembly, the reassembled IP packet length exceeds 65,535. • **Illegal source address 0**—Source address is not a valid address. Invalid addresses are, loopback, broadcast, multicast, and reserved addresses. Source address **0**, however, is allowed to support BOOTP and the destination address **0xffffffff**. • **Illegal destination address 0**—Destination address is not a valid address. The address is reserved. • **TTL zero errors**—Received packet had a time-to-live (TTL) value of 0. • **IP protocol number 0 or 255**—IP protocol is 0 or 255. • **Land attack**—IP source address is the same as the destination address. • **Smurf attack**—Echo request is sent to a directed broadcast address. • **Non-IP packets**—Packet did not conform to the IP standard. • **IP option**—Packet dropped because of a nonallowed IP option. • **Non-IPv4 packets**—Packet was not IPv4. (Only IPv4 is supported.) • **Bad checksum**—Packet had an invalid IP checksum. • **Illegal IP fragment length**—Illegal fragment length. All fragments (other than the last fragment) must have a length that is a multiple of 8 bytes. • **IP fragment overlap**—Fragments have overlapping fragment offsets. • **IP fragment reassembly timeout**—Some of the fragments for an IP packet were not received in time, and the reassembly handler dropped partial fragments.

Table 14: show services stateful-firewall statistics Output Fields *(continued)*

Field Name	Field Description
TCP Errors	TCP protocol errors: • **TCP header length inconsistencies**—Minimum TCP header length is 20 bytes, and the IP packet received does not contain at least 20 bytes. • **Source or destination port number is zero**—TCP source or destination port is zero. • **Illegal sequence number, flags combination**—Dropped because of TCP errors, such as an illegal sequence number, which causes an illogical combination of flags to be set. • **SYN attack (multiple SYN messages seen for the same flow)**—Multiple SYN packets received for the same flow are treated as a SYN attack. The packets might be retransmitted SYN packets and therefore valid, but a large number is cause for concern. • **First packet not SYN**—First packets for a connection are not SYN packets. These packets might originate from previous connections or from someone performing an ACK/FIN scan. • **TCP port scan (Handshake, RST seen from server for SYN)**—In the case of a SYN defender, if an RST (reset) packet is received instead of a SYN/ACK message, someone is probably trying to scan the server. This behavior can result in false alarms if the RST packet is not combined with an intrusion detection service (IDS). • **Bad SYN cookie response**—SYN cookie generates a SYN/ACK message for all incoming SYN packets. If the ACK received for the SYN/ACK message does not match, this counter is incremented.
UDP Errors	UDP protocol errors: • **IP data length less than minimum UDP header length (8 bytes)**—Minimum UDP header length is 8 bytes. The received IP packets contain less than 8 bytes. • **Source or destination port is zero**—UDP source or destination port is **0**. • **UDP port scan (ICMP error seen for UDP flow)**—ICMP error is received for a UDP flow. This could be a genuine UDP flow, but it is counted as an error.
ICMP Errors	ICMP protocol errors: • **IP data length less than minimum ICMP header length (8 bytes)**—ICMP header length is 8 bytes. This counter is incremented when received IP packets contain less than 8 bytes. • **ICMP error length inconsistencies**—Minimum length of an ICMP error packet is 48 bytes, and the maximum length is 576 bytes. This counter is incremented when the received ICMP error falls outside this range. • **Ping duplicate sequence number**—Received ping packet has a duplicate sequence number. • **Ping mismatched sequence number**—Received ping packet has a mismatched sequence number.

Sample Output

show services stateful-firewall statistics extensive

```
user@host> show services stateful-firewall statistics extensive
Interface: ms-1/3/0
  Service set: interface-svc-set
    New flows:
      Accept: 907, Discard: 0, Reject: 0
    Existing flows:
      Accept: 3535, Discard: 0, Reject: 0
    Drops:
      IP option: 0, TCP SYN defense: 0
      NAT ports exhausted: 0
    Errors:
      IP: 0, TCP: 0
      UDP: 0, ICMP: 0
      Non-IP packets: 0, ALG: 0
    IP errors:
      IP packet length inconsistencies: 0
      Minimum IP header length check failures: 0
      Reassembled packet exceeds maximum IP length: 0
      Illegal source address: 0
      Illegal destination address: 0
      TTL zero errors: 0, IP protocol number 0 or 255: 0
      Land attack: 0, Smurf attack: 0
      Non IP packets: 0, IP option: 0
      Non-IPv4 packets: 0, Bad checksum: 0
      Illegal IP fragment length: 0
      IP fragment overlap: 0
      IP fragment reassembly timeout: 0
TCP errors:
      TCP header length inconsistencies: 0
      Source or destination port number is zero: 0
      Illegal sequence number, flags combination: 0
      SYN attack (multiple SYNs seen for the same flow): 0
      First packet not SYN: 0
      TCP port scan (Handshake, RST seen from server for SYN): 0
      Bad SYN cookie response: 0
    UDP errors:
      IP data length less than minimum UDP header length (8 bytes): 0
      Source or destination port is zero: 0
      UDP port scan (ICMP error seen for UDP flow): 0
    ICMP errors:
      IP data length less than minimum ICMP header length (8 bytes): 0
      ICMP error length inconsistencies: 0
      Ping duplicate sequence number: 0
      Ping mismatched sequence number: 0
    ALG drops:
      BOOTP: 0, DCE-RPC: 0, DCE-RPC portmap: 0
      DNS: 0, Exec: 0, FTP: 0
      ICMP: 0
      Login: 0, Netbios: 0, Netshow: 0
      RPC: 0, RPC portmap: 0
      RTSP: 0, Shell: 0
      SNMP: 0, Sqlnet: 0, TFTP: 0
      Traceroute: 0
```


PART 4

Index

- Index on page 237

Index

Symbols

#, comments in configuration statements....xvi
( ), in syntax descriptions....xvi
6rd flows
  statistics....214
< >, in syntax descriptions....xvi
[ ], in configuration statements....xvi
{ }, in configuration statements....xvi
| (pipe), in syntax descriptions....xvi

A

address statement
  NAT....125
    usage guidelines....21
address-allocation statement....125
address-range statement
  NAT....126
allow-overlapping-nat-pools statement....126
application-sets statement
  NAT....127
    usage guidelines....32
applications statement
  NAT....127
    usage guidelines....32

B

basic-nat-pt option
  configuring....53
  example....84
basic-nat44 option
  configuring....37
  example....76
  example, multiple prefixes and address ranges....77
basic-nat66 option
  configuring....41
  example....76
braces, in configuration statements....xvi
brackets
  angle, in syntax descriptions....xvi
  square, in configuration statements....xvi

C

cgn-pic statement....128
  NAT....36
clear services inline nat pool command....202
clear services inline nat statistics command....203
comments, in configuration statements....xvi
configuring dynamic source address and static destination address translation (IPv6 to IPV4)....65
configuring dynamic source address and static destination address translation (IPv6-to-IPv4) example....81
configuring NAT-PT with DNS application-level gateways....58, 62
  example....86
conventions
  text and syntax....xv
curly braces, in configuration statements....xvi
customer support....xvii
  contacting JTAC....xvii

D

destination NAT
  configuring....51, 66, 69
  example....80
destination-address statement
  NAT....128
    usage guidelines....32
destination-address-range statement
  NAT....129
    usage guidelines....32
destination-pool statement....129
  usage guidelines....33
destination-port range statement
  NAT....130
destination-prefix statement....130
destination-prefix-list statement
  NAT....131
destined-port statement
  NAT....131
deterministic-port-block-allocation statement....132
dnat-44 option
  example....80
  usage guidelines....51, 66, 69
documentation
  comments on....xvii
DS-Lite flows
  statistics....214

ds-lite statement....182
usage guidelines....157
dynamic address-only source translation
configuring....48
example....79
dynamic NAT
configuring....48
example....79
dynamic source address and static destination address translation
configuring....65
example....81
dynamic-nat44 option
example....79
usage guidelines....48

F

font conventions....xv
from statement
NAT....133
usage guidelines....30, 32

H

hint statement....134

I

inline NAT
statistics, displaying....204, 205
IPv4
napt-44 option....44
napt-44 option, example....78
translation type
basic-nat-pt option....53
basic-nat44 option....37
basic-nat66 option....41
IPv4 dynamic source translation
configuring....44
example....78
IPv6
napt-66 option....47
IPv6 dynamic source translation
configuring....47
ipv6-multicast-interfaces statement....135
IPv6-to-IPv4 address translation
configuring....65
example....81

M

manuals
comments on....xvii
match-direction statement
NAT....135
usage guidelines....30

N

NAPT
configuring....44, 47
IPv4....44
IPv6....47
port block allocation....25
napt-44 option
example....78
usage guidelines....44
napt-66 option
usage guidelines....47
napt-pt option
example....86
usage guidelines....58, 62
NAT
action statements....33
address configuration....21
applications....32
destination NAT....51, 66
example....80
dynamic address- only source translation....48
dynamic address-only source translation....79
dynamic NAT....48
example....79
dynamic source address and static destination address translation (IPv6 to IPV4)....65
dynamic source address and static destination address translation (IPv6-to-IPv4) example....81
dynamic source translation....44, 47
dynamic source translation, example....78
ipv6-multicast-interfaces information, displaying....206
mapping information, displaying....211
match conditions....32
NAT-PT....58, 62
NAT-PT example....86
rule sets....35
service sets....36
stateful NAT (IPv6 to IPV4)....65
stateful NAT (IPv6-to-IPv4) example....81

static destination address translation.....51, 66
example.....80
status information, displaying.....208
twice NAT
description.....9
nat-type statement.....136
Network Address Port Translation (NAPT)
example.....78
IPv4 example.....78
no-translation statement.....136
usage guidelines.....33

O

overload-pool statement.....137
usage guidelines.....33
overload-prefix statement.....137
usage guidelines.....33

P

parentheses, in syntax descriptions.....xvi
pgcp statement
NAT.....138
pool statement.....139
usage guidelines.....21
port block allocation.....25
deterministic.....25
algorithms.....25
configuring.....72
secured.....25
configuring.....70
port forwarding
configuring.....69
dnat-44.....66
static destination address translation.....66
without destination address translation.....69
port forwarding without static destination address translation
configuring.....69
port statement
NAT.....140
usage guidelines.....21
port-forwarding
example.....60, 106
port-forwarding statement
destined-port statement.....131
NAT.....141
translated-port statement.....149
port-forwarding-mappings statement.....141
ports-per-session statement.....142

R

random-allocation statement.....140
remotely-controlled statement.....142
rule statement
NAT.....143
usage guidelines.....30
softwire.....159, 183
rule-set statement
NAT.....144
usage guidelines.....35
softwire.....183

S

secured-port-block-allocation statement.....145
service-set statement
NAT.....36
services statement
NAT.....144
show services inline nat pool command.....204
show services inline nat statistics command.....205
show services nat ipv6-multicast-interfaces command.....206
show services nat mapping command.....211
show services nat pool command.....208
show services softwire command.....213
show services softwire flows command.....214
show services softwire statistics command.....217
show services stateful-firewall conversations command.....221
show services stateful-firewall flows command.....225
show services stateful-firewall statistics command.....230
softwire flows
statistics.....214
softwire-concentrator statement.....184
softwire-rules statement.....184
source-address statement
NAT.....146
usage guidelines.....32
source-address-range statement
NAT.....146
usage guidelines.....32
source-pool statement.....147
usage guidelines.....33
source-prefix statement.....147
source-prefix-list statement
NAT.....148

stateful firewall
conversations
displaying....221
flows
displaying....225
statistics
displaying....230
stateful NAT
configuring....65
example....81
stateful-nat64 option
example....81
usage guidelines....65
static destination address translation
configuring....51, 66
example....80
support, technical *See* technical support
syntax conventions....xv
syslog statement
NAT....148
usage guidelines....33

T

technical support
contacting JTAC....xvii
term statement
NAT....150
usage guidelines....30
then statement
NAT....151
usage guidelines....30
topic1....213
topic2
sub-topic....213
translated statement....152
usage guidelines....33
translated-port statement
NAT....149
translation-type statement....153
basic-nat-pt option....53
basic-nat44 option....37
basic-nat66 option....41
dnat-44 option, configuring....51, 66
dnat-44 option, example....80
dynamic-nat44, configuring....48
dynamic-nat44, example....79
napt-44 option, configuring....44
napt-44 option, example....78
napt-66 option, configuring....47
napt-pt option, configuring....58, 62
napt-pt option, example....86
stateful-nat64 option, configuring....65
stateful-nat64 option, example....81
usage guidelines....33
transport statement
NAT....156
twice NAT....9
twice-napt-44 option
example....60, 106

V

v6rd statement....185
usage guidelines....158