Introduction to Service PICs

Published: 2012-02-28

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

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

List of Figures

List of Tables

About the Documentation

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

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 xi 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 xi 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

CHAPTER 1

Service PIC Types and Properties

Services PIC Types

Services interfaces enable you to add services to your network incrementally. The Juniper Networks Junos OS supports the following services PICs:

- Adaptive services interfaces (Adaptive Services [AS] PICs and Multiservices PICs)—Enable you to perform multiple services on the same PIC by configuring a set of services and applications. The AS and Multiservices PICs offer a special range of services you configure in one or more service sets: stateful firewalls, Network Address Translation (NAT), intrusion detection service (IDS), class-of-service functionality, and IP Security (IPsec). You can also configure voice services and Layer 2 Tunneling Protocol (L2TP) services. For more information about these services, see "Adaptive Services Overview" on page 4.

  NOTE: **On Juniper Networks MX Series 3D Universal Edge Routers, the Multiservices DPC provides essentially the same capabilities as the Multiservices PIC. The interfaces on both platforms are configured in the same way.**

- ES PIC—Provides a security suite for the IP version 4 (IPv4) and IP version 6 (IPv6) network layers. The suite provides functionality such as authentication of origin, data integrity, confidentiality, replay protection, and nonrepudiation of source. It also defines mechanisms for key generation and exchange, management of security associations, and support for digital certificates. For more information about encryption interfaces, see Configuring Encryption Interfaces.
- Monitoring Services PICs—Enable you to monitor traffic flow and export the monitored traffic. Monitoring traffic allows you to perform the following tasks:
  - Gather and export detailed information about IPv4 traffic flows between source and destination nodes in your network.
  - Sample all incoming IPv4 traffic on the monitoring interface and present the data in cflowd record format.

- Perform discard accounting on an incoming traffic flow.
- Encrypt or tunnel outgoing cflowd records, intercepted IPv4 traffic, or both.
- Direct filtered traffic to different packet analyzers and present the data in its original format.

For more information about flow monitoring interfaces, see Flow Monitoring.

- Multilink Services and Link Services PICs—Enable you to split, recombine, and sequence datagrams across multiple logical data links. The goal of multilink operation is to coordinate multiple independent links between a fixed pair of systems, providing a virtual link with greater bandwidth than any of the members. The Junos OS supports two services PICs based on the Multilink Protocol: the Multilink Services PIC and the Link Services PIC. For more information about multilink and link services interfaces, see Link and Multilink Properties.
- Tunnel Services PIC—By encapsulating arbitrary packets inside a transport protocol, provides a private, secure path through an otherwise public network. Tunnels connect discontinuous subnetworks and enable encryption interfaces, virtual private networks (VPNs), and MPLS. For more information about tunnel interfaces, see Tunnel Properties.

Adaptive Services Overview

The Adaptive Services (AS) and MultiServices PICs provide *adaptive services interfaces*, which allow you to coordinate multiple services on a single PIC by configuring a set of services and applications. The AS and MultiServices PICs offers a special range of services you configure in one or more service sets.

The AS PIC is available in two versions that differ in memory size:

- The Adaptive Services II PIC with 512 MB of memory is supported on all Juniper Networks M Series and T Series routers, including the M320 router.
- The Adaptive Services PIC with 256 megabytes (MB) of memory is supported on all M Series routers except the M320 router.

The M7i router includes the Adaptive Services Module (ASM), an integrated version of the AS PIC as an optional component, which offers all the features of the standalone version at a reduced bandwidth.

NOTE: To take advantage of the features available on the AS PIC, you must install it in an Enhanced Flexible PIC Concentrator (FPC) in an M Series router equipped with an Internet Processor II application-specific integrated circuit (ASIC), or a similarly equipped T Series router. To find out whether your router hardware is suitably equipped, use the **show chassis hardware** command. For more information, see the *Junos OS System Basics and Services Command Reference*.

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

The MultiServices DPC is available for MX Series routers; it includes a subset of the functionality supported on the MultiServices PIC. Currently the MultiServices DPC supports the following Layer 3 services: stateful firewall, NAT, IDS, IPsec, active flow monitoring, RPM, and generic routing encapsulation (GRE) tunnels (including GRE key and fragmentation); it also supports graceful Routing Engine switchover (GRES) and Dynamic Applicaton Awareness for Junos OS. For more information about supported packages, see "Enabling Service Packages" on page 13.

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

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

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

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

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

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

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

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

- Application protocols definition—Allows you to configure properties of application protocols that are subject to processing by router services, and group the application definitions into application sets.
- Service-set definition—Allows you to configure combinations of directional rules and default settings that control the behavior of each service in the service set.

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

CHAPTER 2

Supported Platforms

Supported Platforms

For information about which platforms support Adaptive Services and MultiServices PICs and their features, see "Enabling Service Packages" on page 13.

For information about PIC support on a specific Juniper Networks M Series Multiservice Edge Router or T Series Core Router, see the appropriate *PIC Guide* for the platform.

For information about MS-DPC support on a specific MX Series router, see the appropriate *DPC Guide* for the platform.

For information about services supported on Juniper Networks SRX Series Services Gateways and J Series Services Routers, see the *Junos OS Feature Support Reference for SRX Series and J Series Devices*.

CHAPTER 3

Packet Flow Through Service PICs

Packet Flow Through the Adaptive Services or Multiservices PIC

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

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

The packet flow is as follows, graphically displayed in Figure 1 on page 10. (You can configure a service set as either an interface service set or a next-hop service set.)

1. Packets enter the router on the inbound interface.
2. A policer, filter, service filter, service set, postservice filter, and input forwarding-table filter are applied sequentially to the traffic; these are all optional items in the configuration. If an interface service set is applied, the packets are forwarded to the AS or MultiServices PIC for services processing and then sent back to the Packet Forwarding Engine; if a service filter is also applied, only packets matching the service filter are sent to the PIC. The optional postservice filter is applied and postprocessing takes place.
3. A next-hop service set can be applied to the VPN routing and forwarding (VRF) table or to **inet.0**. If it is applied, packets are sent to the PIC for services processing and sent back to the Packet Forwarding Engine.

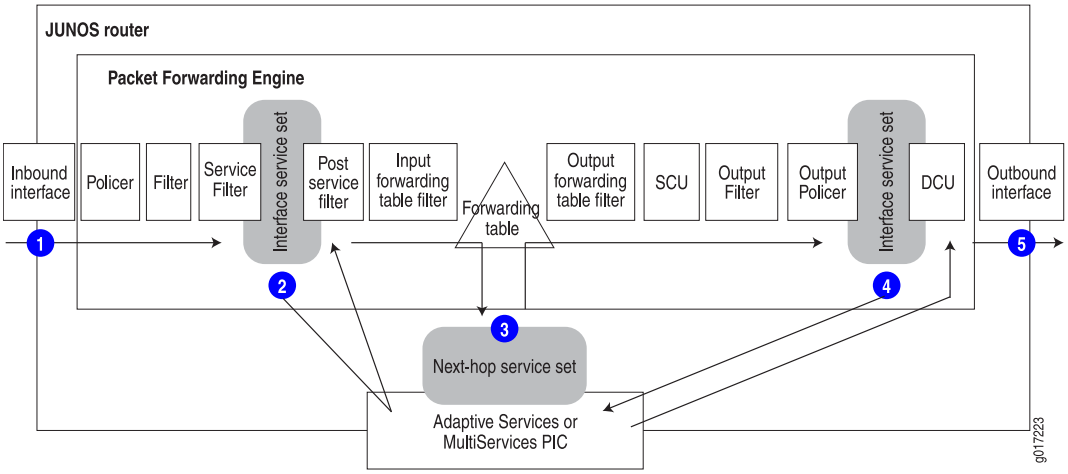
   NOTE: **For NAT, the next-hop service set can only be applied to the VRF table. For all other services, the next-hop service set can be applied to either the VRF table or to inet.0.**

4. On the output interface, an output filter, output policer, and interface service set can be applied sequentially to the traffic if you have configured any of these items. If an

interface service set is applied, the traffic is forwarded to the PIC for processing and sent back to the Packet Forwarding Engine, which then forwards the traffic.

5. Packets exit the router.

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



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

PART 2

Configuration

CHAPTER 4

Configuration Task for Service Packages

Enabling Service Packages

For AS PICs, Multiservices PICs, Multiservices DPCs, and the internal Adaptive Services Module (ASM) in the M7i router, there are two service packages: Layer 2 and Layer 3. Both service packages are supported on all adaptive services interfaces, but you can enable only one service package per PIC, with the exception of a combined package supported on the ASM. On a single router, you can enable both service packages by installing two or more PICs on the platform.

NOTE: Graceful Routing Engine switchover (GRES) is automatically enabled on all services PICs and DPCs except the ES PIC. It is supported on all M Series, MX Series, and T Series routers except for TX Matrix routers. Layer 3 services should retain state after switchover, but Layer 2 services will restart. For IPsec services, Internet Key Exchange (IKE) negotiations are not stored and must be restarted after switchover. For more information about GRES, see the *Junos OS High Availability Configuration Guide*.

You enable service packages per PIC, not per port. For example, if you configure the Layer 2 service package, the entire PIC uses the configured package. To enable a service package, include the service-package statement at the **[edit chassis fpc *slot-number* pic *pic-number* adaptive-services]** hierarchy level, and specify **layer-2** or **layer-3**:

```
[edit chassis fpc slot-number pic pic-number adaptive-services]
service-package (layer-2 | layer-3);
```

To determine which package an AS PIC supports, issue the **show chassis hardware** command: if the PIC supports the Layer 2 package, it is listed as **Link Services II**, and if it supports the Layer 3 package, it is listed as **Adaptive Services II**. To determine which package a Multiservices PIC supports, issue the **show chassis pic fpc-slot *slot-number* pic-slot *slot-number*** command. The **Package** field displays the value **Layer-2** or **Layer-3**.

NOTE: The ASM has a default option (layer-2-3) that combines the features available in the Layer 2 and Layer 3 service packages.

After you commit a change in the service package, the PIC is taken offline and then brought back online immediately. You do not need to manually take the PIC offline and online.

NOTE: Changing the service package causes all state information associated with the previous service package to be lost. You should change the service package only when there is no active traffic going to the PIC.

The services supported in each package differ by PIC and platform type. Table 3 on page 14 lists the services supported within each service package for each PIC and platform. For information about services supported on SRX Series Services Gateways and J Series Services Routers, see the *Junos OS Feature Support Reference for SRX Series and J Series Devices*.

On the AS and Multiservices PICs, *link services* support includes Junos OS CoS components, LFI (FRF.12), MLFR end-to-end (FRF.15), MLFR UNI NNI (FRF.16), MLPPP (RFC 1990), and multiclass MLPPP. For more information, see "Layer 2 Service Package Capabilities and Interfaces" on page 16 and Layer 2 Service Package Capabilities and Interfaces.

NOTE: The AS PIC II for Layer 2 Service is dedicated to supporting the Layer 2 service package only.

For additional information about Layer 3 services, see the *Junos OS Feature Guides*.

Table 3: AS and Multiservices PIC Services by Service Package, PIC, and Platform

Services	ASM	AS/AS2 PICs and Multiservices PICs	AS/AS2 and Multiservices PICs	AS2 and Multiservices PICs	AS2 and Multiservices PICs
Layer 2 Service Package (Only)	M7i	M7i, M10i, and M20	M40e and M120	M320, T320, and T640	TX Matrix
Link Services:					
• Link services	Yes	Yes	Yes	Yes	No
• Multiclass MLPPP	Yes	Yes	Yes	Yes	No
Voice Services:					
• CRTP and LFI	Yes	Yes	Yes	Yes	No
• CRTP and MLPPP	Yes	Yes	Yes	Yes	No
• CRTP over PPP (without MLPPP)	Yes	Yes	Yes	Yes	No

Table 3: AS and Multiservices PIC Services by Service Package, PIC, and Platform *(continued)*

Services	ASM	AS/AS2 PICs and Multiservices PICs	AS/AS2 and Multiservices PICs	AS2 and Multiservices PICs	AS2 and Multiservices PICs
Layer 3 Service Package (Only)	**M7i**	**M7i, M10i, and M20**	**M40e and M120**	**M320, T320, and T640**	**TX Matrix**
Security Services:					
• CoS	Yes	Yes	Yes	Yes	No
• Intrusion detection system (IDS)	Yes	Yes	Yes	Yes	No
• IPsec	Yes	Yes	Yes	Yes	No
• NAT	Yes	Yes	Yes	Yes	No
• Stateful firewall	Yes	Yes	Yes	Yes	No
Accounting Services:					
• Active monitoring	Yes	Yes	Yes	Yes	Yes
• Dynamic flow capture (Multiservices 400 PIC only)	No	No	No	Yes	No
• Flow-tap	Yes	Yes	Yes (M40e only)	Yes	No
• Passive monitoring (Multiservices 400 PIC only)	No	Yes	Yes (M40e only)	Yes	No
• Port mirroring	Yes	Yes	Yes	Yes	Yes
LNS Services:					
• L2TP LNS	Yes	Yes (M7i and M10i only)	Yes (M120 only)	No	No
Voice Services:					
• BGF	Yes	Yes	Yes	Yes	No
Layer 2 and Layer 3 Service Package (Common Features)	**M7i**	**M7i, M10i, and M20**	**M40e and M120**	**M320, T320, and T640**	**TX Matrix**
RPM Services:					
• RPM probe timestamping	Yes	Yes	Yes	Yes	No

Table 3: AS and Multiservices PIC Services by Service Package, PIC, and Platform *(continued)*

Services	ASM	AS/AS2 PICs and Multiservices PICs	AS/AS2 and Multiservices PICs	AS2 and Multiservices PICs	AS2 and Multiservices PICs
Tunnel Services:					
• GRE (**gr-*fpc/pic/port***)	Yes	Yes	Yes	Yes	Yes
• GRE fragmentation (**clear-dont-fragment-bit**)	Yes	Yes	Yes	No	No
• GRE key	Yes	Yes	Yes	Yes	No
• IP-IP tunnels (**ip-*fpc/pic/port***)	Yes	Yes	Yes	Yes	Yes
• Logical tunnels (**lt-*fpc/pic/port***)	No	No	No	No	No
• Multicast tunnels (**mt-*fpc/pic/port***)	Yes	Yes	Yes	Yes	Yes
• PIM de-encapsulation (**pd-*fpc/pic/port***)	Yes	Yes	Yes	Yes	Yes
• PIM encapsulation (**pe-*fpc/pic/port***)	Yes	Yes	Yes	Yes	Yes
• Virtual tunnels (**vt-*fpc/pic/port***)	Yes	Yes	Yes	Yes	Yes

Layer 2 Service Package Capabilities and Interfaces

When you enable the Layer 2 service package, you can configure link services. On the AS and Multiservices PICs and the ASM, link services include support for the following:

- Junos CoS components—Layer 2 Service Package Capabilities and Interfaces describes how the Junos CoS components work on link services IQ (**lsq**) interfaces. For detailed information about Junos CoS components, see the *Junos OS Class of Service Configuration Guide*.
- LFI on Frame Relay links using FRF.12 end-to-end fragmentation—The standard for FRF.12 is defined in the specification FRF.12, *Frame Relay Fragmentation Implementation Agreement*.
- LFI on MLPPP links.
- MLFR UNI NNI (FRF.16)—The standard for FRF.16 is defined in the specification FRF.16.1, *Multilink Frame Relay UNI/NNI Implementation Agreement*.
- MLPPP (RFC 1990)
- MLFR end-to-end (FRF.15)

For the LSQ interface on the AS and Multiservices PICs, the configuration syntax is almost the same as for Multilink and Link Services PICs. The primary difference is the use of the

interface-type descriptor **lsq** instead of **ml** or **ls**. When you enable the Layer 2 service package, the following interfaces are automatically created:

gr-*fpc/pic/port*
ip-*fpc/pic/port*
lsq-*fpc/pic/port*
lsq-*fpc/pic/port*:0
...
lsq-*fpc/pic/port*:*N*
mt-*fpc/pic/port*
pd-*fpc/pic/port*
pe-*fpc/pic/port*
sp-*fpc/pic/port*
vt-*fpc/pic/port*

Interface types **gr**, **ip**, **mt**, **pd**, **pe**, and **vt** are standard tunnel interfaces that are available on the AS and Multiservices PICs whether you enable the Layer 2 or the Layer 3 service package. These tunnel interfaces function the same way for both service packages, except that the Layer 2 service package does not support some tunnel functions, as shown in Table 3 on page 14.

Interface type **lsq-*fpc/pic/port*** is the physical link services IQ (**lsq**) interface. Interface types **lsq-*fpc/pic/port*:0** through **lsq-*fpc/pic/port*:*N*** represent FRF.16 bundles. These interface types are created when you include the **mlfr-uni-nni-bundles** statement at the **[edit chassis fpc *slot-number* pic *pic-number*]** option. For more information, see Layer 2 Service Package Capabilities and Interfaces and Link and Multilink Properties.

NOTE: Interface type **sp** is created because it is needed by the Junos OS. For the Layer 2 service package, the **sp** interface is not configurable, but you should not disable it.

CHAPTER 5

Configuration Task for Services

Services Configuration Procedure

You follow these general steps to configure services:

1. Define application objects by configuring statements at the **[edit applications]** hierarchy level.
2. Define service rules by configuring statements at the **[edit services (ids | ipsec-vpn | nat | stateful-firewall) rule]** hierarchy level.
3. Group the service rules by configuring the **rule-set** statement at the **[edit services (ids | ipsec-vpn | nat | stateful-firewall)]** hierarchy level.
4. Group service rule sets under a service-set definition by configuring the **service-set** statement at the **[edit services]** hierarchy level.
5. Apply the service set on an interface by including the **service-set** statement at the **[edit interfaces *interface-name* unit *logical-unit-number* family inet service (input | output)]** hierarchy level. Alternatively, you can configure logical interfaces as a next-hop destination by including the **next-hop-service** statement at the **[edit services service-set *service-set-name*]** hierarchy level.

NOTE: You can configure IDS, NAT, and stateful firewall service rules within the same service set. You must configure IPsec services in a separate service set, although you can apply both service sets to the same PIC.

CHAPTER 6

Examples

Example: Service Interfaces Configuration

The following configuration includes all the items necessary to configure services on an interface. For examples showing individual service configurations, see the chapters that describe each service in detail.

```
[edit]
interfaces {
   fe-0/1/0 {
      unit 0 {
         family inet {
            service {
               input {
                  service-set Firewall-Set;
               }
               output {
                  service-set Firewall-Set;
               }
            }
            address 10.1.3.2/24;
         }
      }
   }
   fe-0/1/1 {
      unit 0 {
         family inet {
            filter {
               input Sample;
            }
            address 172.16.1.2/24;
         }
      }
   }
```

```
    sp-1/0/0 {
      unit 0 {
        family inet {
          address 172.16.1.3/24 {
          }
        }
      }
    }
}
forwarding-options {
    sampling {
      input {
        family inet {
          rate 1;
        }
      }
      output {
        cflowd 10.1.3.1 {
          port 2055;
          version 5;
        }
        flow-inactive-timeout 15;
        flow-active-timeout 60;
        interface sp-1/0/0 {
          engine-id 1;
          engine-type 136;
          source-address 10.1.3.2;
        }
      }
    }
}
firewall {
    filter Sample {
      term Sample {
        then {
          count Sample;
          sample;
          accept;
        }
      }
    }
}
services {
    stateful-firewall {
      rule Rule1 {
        match-direction input;
        term 1 {
          from {
            application-sets Applications;
          }
          then {
            accept;
          }
        }
        term accept {
          then {
```

```
            accept;
          }
        }
      }
      rule Rule2 {
        match-direction output;
        term Local {
          from {
            source-address {
              10.1.3.2/32;
            }
          }
          then {
            accept;
          }
        }
      }
    }
    ids {
      rule Attacks {
        match-direction output;
        term Match {
          from {
            application-sets Applications;
          }
          then {
            logging {
              syslog;
            }
          }
        }
      }
    }
    nat {
      pool public {
        address-range low 172.16.2.1 high 172.16.2.32;
        port automatic;
      }
      rule Private-Public {
        match-direction input;
        term Translate {
          then {
            translated {
              source-pool public;
              translation-type source dynamic;
            }
          }
        }
      }
    }
    service-set Firewall-Set {
      stateful-firewall-rules Rule1;
      stateful-firewall-rules Rule2;
      nat-rules Private-Public;
      ids-rules Attacks;
      interface-service {
```

```
        service-interface sp-1/0/0;
      }
    }
  }
  applications {
    application ICMP {
      application-protocol icmp;
    }
    application FTP {
      application-protocol ftp;
      destination-port ftp;
    }
    application-set Applications {
      application ICMP;
      application FTP;
    }
  }
```

Example: VPN Routing and Forwarding (VRF) and Service Configuration

The following example combines VPN routing and forwarding (VRF) and services configuration:

```
[edit policy-options]
policy-statement test-policy {
  term t1 {
    then reject;
  }
}
[edit routing-instances]
test {
  interface ge-0/2/0.0;
  interface sp-1/3/0.20;
  instance-type vrf;
  route-distinguisher 10.58.255.1:37;
  vrf-import test-policy;
  vrf-export test-policy;
  routing-options {
    static {
      route 0.0.0.0/0 next-table inet.0;
    }
  }
}
[edit interfaces]
ge-0/2/0 {
  unit 0 {
    family inet {
      service {
        input service-set nat-me;
        output service-set nat-me;
      }
    }
  }
}
sp-1/3/0 {
```

```
    unit 0 {
      family inet;
    }
    unit 20 {
      family inet;
      service-domain inside;
    }
    unit 21 {
      family inet;
      service-domain outside;
    }
    [edit services]
    stateful-firewall {
      rule allow-any-input {
        match-direction input;
        term t1 {
          then accept;
        }
      }
    }
    nat {
      pool hide-pool {
        address 10.58.16.100;
        port automatic;
      }
      rule hide-all-input {
        match-direction input;
        term t1 {
          then {
            translated {
              source-pool hide-pool;
              translation-type source dynamic;
            }
          }
        }
      }
    }
    service-set nat-me {
      stateful-firewall-rules allow-any-input;
      nat-rules hide-all-input;
      interface-service {
        service-interface sp-1/3/0.20;
      }
    }
  }
```

Example: Dynamic Source NAT as a Next-Hop Service

The following example shows dynamic-source NAT applied as a next-hop service:

```
[edit interfaces]
ge-0/2/0 {
  unit 0 {
    family mpls;
  }
```

```
}
sp-1/3/0 {
  unit 0 {
    family inet;
  }
  unit 20 {
    family inet;
  }
  unit 32 {
    family inet;
  }
}
[edit routing-instances]
protected-domain {
  interface ge-0/2/0.0;
  interface sp-1/3/0.20;
  instance-type vrf;
  route-distinguisher 10.58.255.17:37;
  vrf-import protected-domain-policy;
  vrf-export protected-domain-policy;
  routing-options {
    static {
      route 0.0.0.0/0 next-hop sp-1/3/0.20;
    }
  }
}
[edit policy-options]
policy-statement protected-domain-policy {
  term t1 {
    then reject;
  }
}
[edit services]
stateful-firewall {
  rule allow-all {
    match-direction input;
    term t1 {
      then {
        accept;
      }
    }
  }
}
nat {
  pool my-pool {
    address 10.58.16.100;
    port automatic;
  }
  rule hide-all {
    match-direction input;
    term t1 {
      then {
        translated {
          source-pool my-pool;
          translation-type source dynamic;
        }
```

```
        }
      }
    }
  }
  service-set null-sfw-with-nat {
    stateful-firewall-rules allow-all;
    nat-rules hide-all;
    next-hop-service {
      inside-service-interface sp-1/3/0.20;
      outside-service-interface sp-1/3/0.32;
    }
  }
```

Example: 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: BOOTP and Broadcast Addresses

The following example supports Bootstrap Protocol (BOOTP) and broadcast addresses:

```
[edit applications]
application bootp {
  application-protocol bootp;
  protocol udp;
  destination-port 67;
}
[edit services]
stateful-firewall bootp-support {
  rule bootp-allow {
    direction input;
    term bootp-allow {
      from {
        destination-address {
          any-unicast;
          255.255.255.255;
        }
        application bootp;
      }
      then {
        accept;
      }
    }
  }
}
```

PART 3

Index

Index