

Implementation of IBM j-type Ethernet Appliances



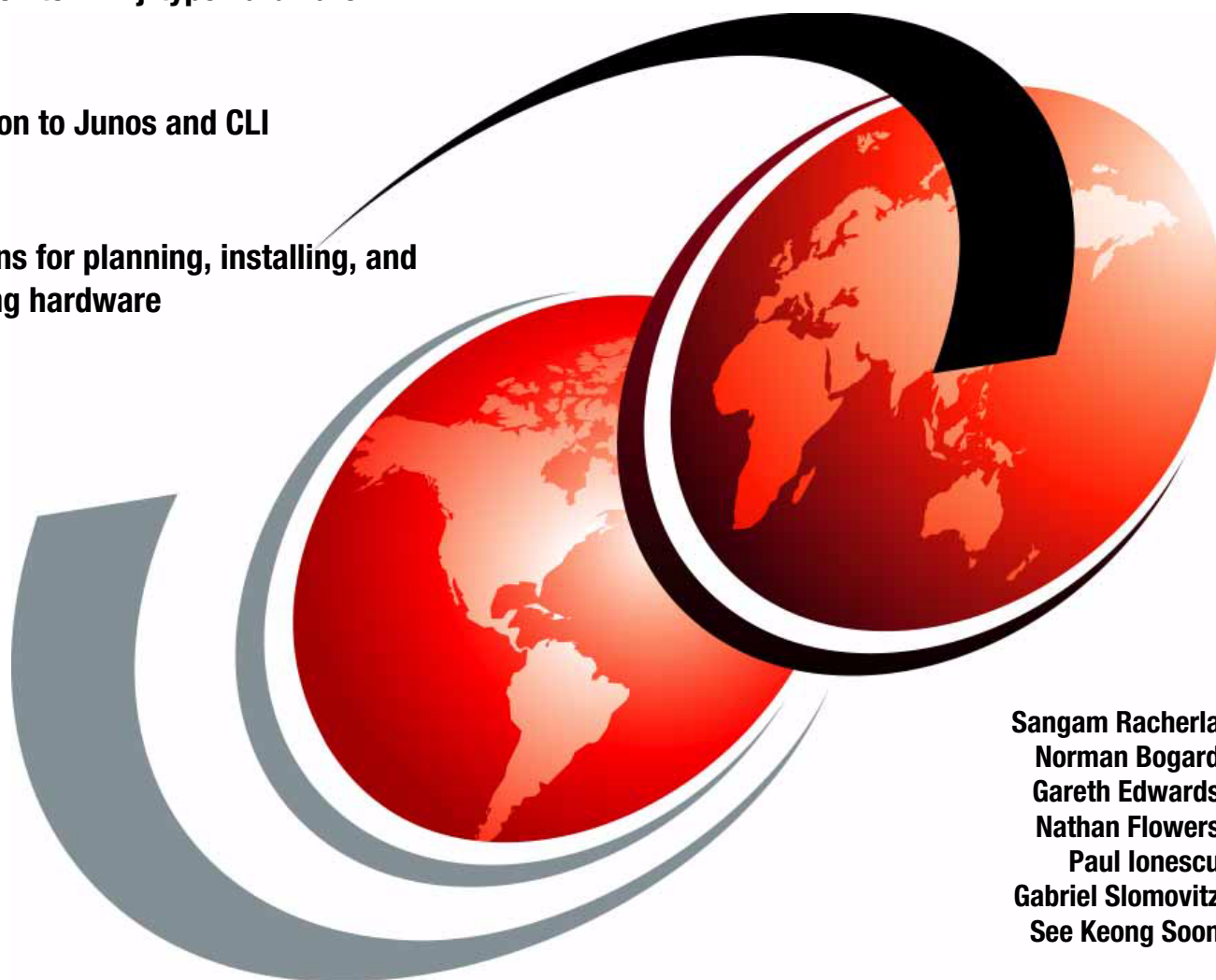
Introduction to IBM j-type hardware



Introduction to Junos and CLI



Instructions for planning, installing, and
configuring hardware



Sangam Racherla
Norman Bogard
Gareth Edwards
Nathan Flowers
Paul Ionescu
Gabriel Slomovitz
See Keong Soon

Redbooks



International Technical Support Organization

Implementation of IBM j-type Ethernet Appliances

February 2011

Note: Before using this information and the product it supports, read the information in “Notices” on page ix.

First Edition (February 2011)

This edition applies to IBM j-type Ethernet Appliance operating on Junos Software Vervion 10.1

© Copyright International Business Machines Corporation 2011. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	ix
Trademarks	x
 Preface	xi
The team who wrote this book	xi
Now you can become a published author, too!	xiv
Comments welcome	xiv
Stay connected to IBM Redbooks	xiv
 Chapter 1. Fundamentals of Ethernet networking	1
1.1 Open System Interconnect (OSI) Model	2
1.1.1 OSI layer descriptions	3
1.2 General networking concepts	3
1.2.1 Local Area Network	3
1.2.2 Wide Area Network	3
1.3 Layer 1 networking concepts and terminology	4
1.3.1 Ethernet cabling	4
1.3.2 Layer 1 components	11
1.3.3 Network segments	12
1.3.4 Physical configuration parameters	13
1.3.5 Power over Ethernet (PoE)	14
1.4 Layer 2 networking concepts and terminology	16
1.4.1 Frame structure	16
1.4.2 Jumbo frames	17
1.4.3 Interframe gap	17
1.4.4 Media Access Control addresses	17
1.4.5 Bridging or Layer 2 switching	20
1.4.6 Virtual Local Area Network	23
1.4.7 Interface VLAN operation modes	26
1.4.8 Link aggregation	27
1.4.9 Spanning Tree Protocol	27
1.4.10 Link Layer Discovery Protocol	30
1.4.11 LLDP TLVs	30
1.5 Layer 3 networking concepts and terminology	32
1.5.1 IP routing	33
1.5.2 Address Resolution Protocol	34
1.5.3 IPv4 addressing	34
1.5.4 IPv6 addressing	36
1.6 Firewalls	38
1.6.1 Personal firewalls	38
1.6.2 Network firewalls	38
 Chapter 2. Introduction to the IBM j-type Ethernet appliances	41
2.1 IBM Ethernet Appliance J34S and IBM Ethernet Appliance J36S	42
2.2 IBM Ethernet Appliance J56S and IBM Ethernet Appliance J58S	54
2.3 Junos operating system	66
2.3.1 A common network operating system	66
2.4 More information	68

Chapter 3. Initial hardware planning of IBM j-type appliances	71
3.1 Installation overview	72
3.2 Power considerations	72
3.2.1 AC power circuit breaker requirements for the IBM j-type appliances	72
3.2.2 Power specifications and requirements for the IBM Ethernet Appliance J34S and J36S	72
3.2.3 Power specifications and requirements for the IBM Ethernet Appliance J56S and J58S	77
3.3 Cabling	83
3.3.1 Cables connecting to management devices	84
3.4 Cooling system	87
3.4.1 IBM Ethernet Appliance J34S cooling	87
3.4.2 IBM Ethernet Appliance J36S cooling	89
3.4.3 IBM Ethernet Appliance J56S cooling	90
3.4.4 J58S cooling system description	92
3.5 Racks	95
3.5.1 Cabinet airflow requirements	95
3.5.2 Clearance for maintenance	96
3.6 More information	96
Chapter 4. Junos fundamentals	99
4.1 Junos overview	100
4.2 Junos architecture	101
4.3 Routing Engine	101
4.3.1 Junos software Routing Engine components	101
4.4 Packet Forwarding Engine	103
4.5 Junos software configuration model	103
4.5.1 Active configuration	103
4.5.2 Candidate configuration	103
4.5.3 Configuration history	104
4.6 Junos routing and forwarding tables	104
4.7 User interface options	105
4.8 Introduction to Junos CLI	106
4.8.1 Operational mode	106
4.8.2 Configuration mode	106
4.8.3 More information	107
Chapter 5. Initial configuration	109
5.1 Initial IBM j-type Appliance configuration	110
5.1.1 Configuring the Junos software for the first time on an appliance with a Single Routing Engine	110
5.1.2 Configuring the Junos software for the first time on an appliance with Dual Routing Engines	114
5.1.3 Junos software default settings for appliance security	116
5.1.4 Junos software configuration using the CLI	116
5.1.5 Activation of the Junos software candidate configuration	117
5.1.6 Disk space management for Junos software installation	117
5.1.7 Junos software tools for monitoring the router	117
5.2 More information	118
Chapter 6. User interface	119
6.1 J-Web graphical user interface	120
6.1.1 Logging into J-Web GUI	120
6.1.2 Dashboard tab	125

6.1.3 Configure tab	126
6.1.4 Monitor tab	127
6.1.5 Maintain tab	128
6.1.6 Troubleshoot tab	129
6.2 Junos CLI	130
6.2.1 CLI modes	130
6.2.2 Logging in CLI	130
6.2.3 CLI operational mode	131
6.2.4 CLI configuration mode	139
6.3 More information	148
Chapter 7. Class of service	149
7.1 Class of Service	150
7.2 Junos CoS components	150
7.3 Packet flow	151
7.4 Packet classification	153
7.4.1 Behavior aggregate classification	153
7.4.2 Code point alias	157
7.4.3 Multifield classification	159
7.5 Forwarding classes	161
7.6 Simple filters and policers	162
7.6.1 Simple filters	163
7.6.2 Policing	165
7.6.3 Policing configuration (Two-color marking)	166
7.6.4 Policing example (Interface-based policer)	167
7.7 Rewrite rules	168
7.7.1 Default rewrite rules	168
7.7.2 Rewrite rules configuration	169
7.7.3 Rewrite rules example	170
7.8 Packet's Loss Priority (PLP)	170
7.9 Schedulers	171
7.9.1 Default schedulers	171
7.9.2 Schedulers' configuration	171
7.9.3 Schedulers' example	173
7.10 Red Profiles	175
7.10.1 RED Profile configuration	175
7.10.2 RED Profile example	176
7.11 Case study	179
7.12 More information	185
Chapter 8. Network security	187
8.1 Sample topology	188
8.2 Firewall filters (stateless)	188
8.3 Security zones	195
8.3.1 Security zones and interfaces overview	195
8.3.2 Security zones	196
8.3.3 Host inbound traffic	198
8.3.4 Control inbound traffic based on protocols	199
8.3.5 TCP-Reset parameters	199
8.3.6 Address books and address sets	200
8.3.7 Zone configuration example	201
8.4 Security policies	205
8.4.1 Security policies overview	205

8.4.2	Security Policy Schedulers	208
8.4.3	Security Policy Applications	209
8.4.4	Security policy configuration example	212
8.5	Network Address Translation	218
8.5.1	NAT overview	219
8.5.2	Static NAT	221
8.5.3	Static NAT configuration example	222
8.5.4	Destination NAT	225
8.5.5	Destination NAT configuration example	227
8.5.6	Source NAT	232
8.5.7	Source NAT configuration example	237
8.6	Virtual Private Networks	244
8.6.1	Internet Protocol Security	244
8.6.2	IPsec VPN configuration overview	251
8.6.3	VPN configuration example	256
8.6.4	IPsec dynamic VPN and PKI support	261
8.7	Authentication options	262
8.7.1	User authentication overview	262
8.7.2	Pass-through authentication	262
8.7.3	Web authentication	263
8.7.4	External authentication servers	265
8.7.5	Client groups for firewall authentication	267
8.7.6	User authentication configuration example	267
8.8	Other security	271
8.8.1	Attack detection and prevention	271
8.8.2	Transparent mode	273
Chapter 9.	Advanced configuration	275
9.1	Chassis cluster	276
9.1.1	Chassis cluster overview	276
9.1.2	Example: Active/passive chassis cluster deployment	278
9.1.3	Example: Active/active chassis cluster deployment	293
9.2	Virtual router	315
9.2.1	Virtual router overview	315
9.2.2	Example: Virtual router deployment	315
9.3	More information	325
Chapter 10.	Management and monitoring	327
10.1	Configuring SNMP for network management	328
10.1.1	SNMP architecture	328
10.1.2	Before you begin	330
10.1.3	Configuring SNMP with Quick Configuration	331
10.1.4	Configuring SNMP with a configuration editor	335
10.1.5	Verifying the SNMP configuration	339
10.2	Monitoring the device and routing operations	341
10.2.1	Monitoring overview	341
10.2.2	Monitoring interfaces	343
10.2.3	Monitoring events and alarms	345
10.2.4	Monitoring the system	345
10.2.5	Monitoring NAT	354
10.2.6	Monitoring security features	356
10.2.7	Monitoring IDP	363
10.2.8	Monitoring flow session statistics	364

10.2.9	Monitoring flow gate information	373
10.2.10	Monitoring firewall authentication	373
10.3	Monitoring events and managing system log files	377
10.3.1	System log message terms.	377
10.3.2	System log messages overview	380
10.3.3	Configuring system log messages with a configuration editor	382
10.3.4	Monitoring system log messages with the J-Web event viewer	386
10.4	Configuring and monitoring alarms	388
10.4.1	Alarm terms.	388
10.4.2	Alarm overview	389
10.4.3	Configuring alarms with a configuration editor	394
10.4.4	Checking active alarms.	396
10.5	Additional material.	399
Chapter 11.	Maintenance and analysis	401
11.1	Basic systems functions	402
11.1.1	Rebooting or halting the system	402
11.1.2	Bringing chassis components online and offline	403
11.1.3	Restarting a software process.	404
11.1.4	Managing files.	404
11.1.5	Setting rescue configuration	406
11.1.6	Reverting to rescue configuration	406
11.1.7	Reverting to default factory settings	406
11.1.8	Recovering passwords	407
11.2	Network utilities	408
11.2.1	Ping and traceroute.	408
11.2.2	Telnet, SSH, and FTP clients	409
11.3	Diagnosing tools	410
11.3.1	Show interfaces command	411
11.3.2	Monitor interface command.	414
11.3.3	Monitor traffic command	416
11.3.4	Monitoring system commands	418
11.3.5	Displaying log and trace files	420
11.4	Managing Junos software	420
11.4.1	Understanding software packaging.	421
11.4.2	Understanding recovery software packaging	421
11.4.3	System snapshot	421
11.4.4	Upgrading Junos software	422
11.4.5	Downgrading Junos software	424
11.5	Managing licences	425
11.5.1	Licence key components	425
11.5.2	Generating a license key	425
11.5.3	Managing Junos software licenses	425
11.5.4	Verifying Junos software licenses.	427
11.6	File system overview.	428
11.7	Configuration management.	430
11.7.1	Saving the configuration	430
11.7.2	Returning to the most recently committed configuration.	431
11.7.3	Returning to a previously committed configuration	431
11.7.4	Displaying previous configurations	431
11.7.5	Comparing configuration changes	432
11.7.6	Saving a configuration to a file	433
11.7.7	Loading a configuration from a file	434

11.8 Chassis and interfaces alarms	435
11.8.1 Interface LEDs on J34S and J36S	435
11.8.2 Alarm relay contacts on J56S and J58S	436
11.8.3 Craft Interface LEDs on J56S and J58S	436
11.8.4 Component LEDs on J56S and J58S	437
11.9 More information	437
Related publications	439
IBM Redbooks	439
Other publications	439
Online resources	441
How to get Redbooks	441
Help from IBM	441
Index	443

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:


AIX®

BladeCenter®

IBM®

POWER®

Redbooks®

Redbooks (logo) ®

System p®

System x®

The following terms are trademarks of other companies:

Adobe, the Adobe logo, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Snapshot, and the NetApp logo are trademarks or registered trademarks of NetApp, Inc. in the U.S. and other countries.

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Preface

The data center is the strategic heart of today's high-performance enterprise, unifying the critical systems, applications, and storage services that are needed for business success. A data center solution that is architected by IBM® and Juniper allows companies to centralize their mission-critical infrastructure with a comprehensive solution that combines best-in-class products with well-defined practices that are designed for the enterprise.

IBM j-type data center solutions that run Junos software provide operational agility and efficiency that dramatically simplifies the network and delivers unprecedented savings. This solution enables a network design with fewer devices, interconnections, and network tiers. Beyond the obvious cost advantages, the design enables the following key benefits:

- ▶ Reduces latency
- ▶ Simplifies device management
- ▶ Delivers significant power, cooling, and space savings
- ▶ Eliminates multiple system failure points
- ▶ Performs pervasive security

The high-performance data center is built around IBM j-type e-series switches, m-series routers, and s-series firewalls. It is a new family of powerful products that help shape the next-generation dynamic infrastructure.

IBM j-type s-series Ethernet Appliances perform essential networking security functions and are ready for next-generation data center services and applications. Designed on top of the Junos operating system, s-series Ethernet Appliances provide flexible processing scalability, I/O scalability, network segmentation, and services integration.

This Redbooks® publication covers the following major topics:

- ▶ Introduction to Ethernet fundamentals and IBM j-type s-series Ethernet Appliance hardware.
- ▶ Initial hardware planning and configuration.
- ▶ Other configuration topics include Chassis Cluster, Virtual Router, Network Security features, and Class of Service.
- ▶ Network management features of Junos Software, and Maintenance of the IBM j-type s-series Ethernet Appliance hardware.

This IBM Redbooks publication targets IT professionals who sell, design, or administer IBM j-type networking solutions.

This publication provides information about IBM j-type Ethernet appliances and can be used with the following publications:

- ▶ *IBM j-type Data Center Networking Introduction*, SG24-7820
- ▶ *IBM j-type Ethernet Switches and Routers Implementation*, SG24-7882

The team who wrote this book

This book was produced by a team of specialists from around the world working at the International Technical Support Organization, Raleigh Center.



Figure 1 The Team: Gareth, Gabriel, Soon, Norman, Paul, Nathan, and Sangam

Sangam Racherla is an Information Technology (IT) Specialist and Project Leader working at the International Technical Support Organization (ITSO), San Jose Center. He has a degree in Electronics and Communication Engineering and has ten years of experience in the IT field. He has been with the ITSO for the past seven years and has extensive experience installing and supporting the ITSO lab equipment for various Redbooks publication projects. His areas of expertise include Microsoft® Windows®, Linux®, AIX®, System x®, and System p® servers, and various SAN and storage products.

Norman Bogard is a Senior Technical Sales Specialist with the IBM Advanced Technical Skills Organization. He has supported Ethernet networking since 1987, Storage Area Networks since 1996, and Network Attached Storage since IBM entered into the sector in 2001. He began his career in Information Technology in 1984 with Intel® Corporation and came into IBM through the acquisition of Sequent Computers.

Gareth Edwards is an IT Specialist for STG in IBM Australia specializing in Storage and POWER® systems. He is certified on the IBM Midrange Storage Systems and Cisco Networking. He has 20 years of experience in the IT industry and has been with IBM since 2006. Gareth's current area of expertise includes planning and implementation of midrange and enterprise servers, storage, and storage networks.

Nathan Flowers is a Network Engineer and has been with IBM for 20 years. He started his career in the National Support Center in Atlanta providing dealer support for PC, XT, AT, PS/2s, and Thinkpad systems. After moving to RTP, he joined the Network Hardware Division's development group as an Asynchronous Transfer Mode and Ethernet Network Engineer in 1998. Nathan has had many roles in the networking environment, such as BladeCenter® networking solutions and BladeCenter and System x networking education. He is currently in the SAN Central and Solutions support group providing product engineering and solutions support for the IBM Data Center Networking products. He has a Bachelor of Science degree in Computer Science from Kennesaw State University.

Paul Ionescu is a Senior IT Specialist working for IBM Global Technology Services Romania with more than 10 years of networking experience. He is a Juniper Networks® Certified Internet Expert (JNCIE-M), Cisco Certified Internetworking Expert (CCIE), and Certified Information Systems Security Professional (CISSP). He is currently focused on network engineering projects for the ISP, telco, and banking industries. Paul joined IBM in 2000.

Gabriel Slomovitz is a Network Specialist at IBM Uruguay. He has almost 10 years of experience working in the design and implementation of networking projects. His main specialties include routing, switching, wireless and RFID. He holds an Engineering degree in Telecommunications from Universidad ORT and he is a Cisco Certified Internetwork Expert (CCIE) in Routing & Switching.

See Keong Soon is an Advisory IT Specialist for IBM Global Technology Services in Malaysia. He has eight years of experience and specializes in network and security technologies. Among many previous appointments he held, See Keong served as a Network Lead to implement many large scale network and security projects. His areas of expertise include planning, designing, and implementing various network and security technologies for a wide range of clients.

Thanks to the following people for their contributions to this project:

International Technical Support Organization, Raleigh Center

Jon Tate
Margaret Ticknor
Tamikia Barrow

IBM

Doris Konieczny
Jeffrey Walls
Steve Grillo
Mark Bayus
Doug Vassello
Jason Daniel
Jim Blue
William Champion
George Pappas

Juniper Networks

Greg Basset
Steve Gonzales
Jermey Wallace
John Nishikawa
Charles Goldberg
Chris Rogers
Steven Smith
Ashutosh Thakur
Aditi Sharma
Brad Woodberg
Sean Capshaw

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author - all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:

ibm.com/redbooks

- Send your comments in an email to:

redbooks@us.ibm.com

- Mail your comments to:

IBM Corporation, International Technical Support Organization
Dept. HYTD Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- Find us on Facebook:

<http://www.facebook.com/IBMRedbooks>

- Follow us on Twitter:

<http://twitter.com/ibmredbooks>

- Look for us on LinkedIn:


<http://www.linkedin.com/groups?home=&gid=2130806>

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

<https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm>

- Stay current on recent Redbooks publications with RSS Feeds:

<http://www.redbooks.ibm.com/rss.html>



Fundamentals of Ethernet networking

This chapter explains the fundamental networking concepts and terminology that are related to topics that we cover in this book.

There are many topics in the networking environment that can be discussed. For the scope of this book, in this chapter, we discuss topics in the following areas:

- ▶ Open System Interconnect (OSI) Model
- ▶ General Networking Concepts
- ▶ Layer 1 Concepts and Terminology
- ▶ Layer 2 Concepts and Terminology
- ▶ Layer 3 Concepts and Terminology
- ▶ Firewalls

1.1 Open System Interconnect (OSI) Model

The Open System Interconnect (OSI) Reference Model divides the computer networking architecture into seven layers for a visual representation of the dependencies and interactions of various communication processes as illustrated in Figure 1-1. Processes are grouped according to their function in the model. Processes at each layer are dependant upon the functions of the layer below and provides services for the layer above.

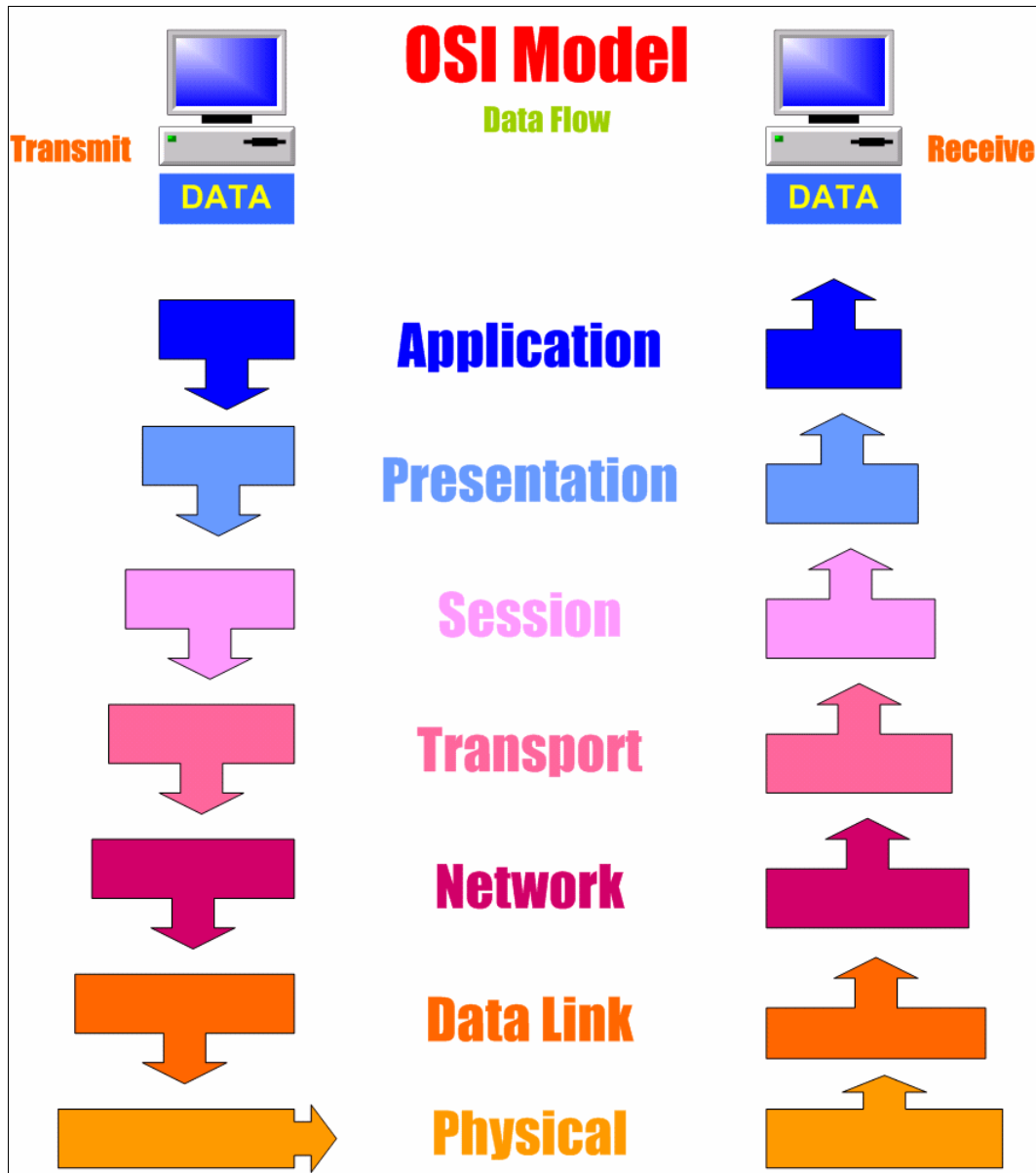


Figure 1-1 OSI Layer

1.1.1 OSI layer descriptions

In this section, we highlight each of the layers with some examples of the functions in each layer:

Layer 7 Application	Provides access to network processes to and from applications, for example: FTP, SMTP, HTTP, SNMP, Telnet, and NTP.
Layer 6 Presentation	Provides data representation and encryption, for example: Secure Sockets Layer (SSL), Transport Layer Security (TLS), and Extensible Markup Language (XML).
Layer 5 Session	Provides interhost communications or process-to-process messages between computers. Manages establishment and termination of sessions, for example: socket addresses.
Layer 4 Transport	Provides end-to-end communications and reliability. Manages flow control, fragmentation, and error control, for example: Transmission Control Protocol (TCP) functions of TCP/IP.
Layer 3 Network	Provides path determination or routing from source to destination, for example: Internet Protocol (IP) functions of TCP/IP.
Layer 2 Data Link	Provides the functional means for data transfer between adjacent nodes in the network, for example: MAC or Physical Addressing.
Layer 1 Physical	Provides electrical and physical parameters (signal and media) for transmission of data, for example: Functions of NICs, hubs, cables, connector types, and signaling parameters.

1.2 General networking concepts

Networks can be categorized by many factors. Two of the most common terms are Local Area Network (LAN) and Wide Area Network (WAN).

1.2.1 Local Area Network

A Local Area Network is a computer network of devices that spans a small geographic area. A LAN can be a small network that is contained within a small office or home and can also be larger in size, such as a large office building or campus. The boundary of the LAN is commonly the point at which the traffic exits the management control of the LAN and enters the management control of an Internet Service Provider (ISP).

1.2.2 Wide Area Network

A Wide Area Network interconnects multiple LANs. The area of a WAN can vary from interconnecting LANs that are near each other, as illustrated in Figure 1-2 on page 4, or can span the world. It can be composed of many technologies with a wide variety of capabilities and features.

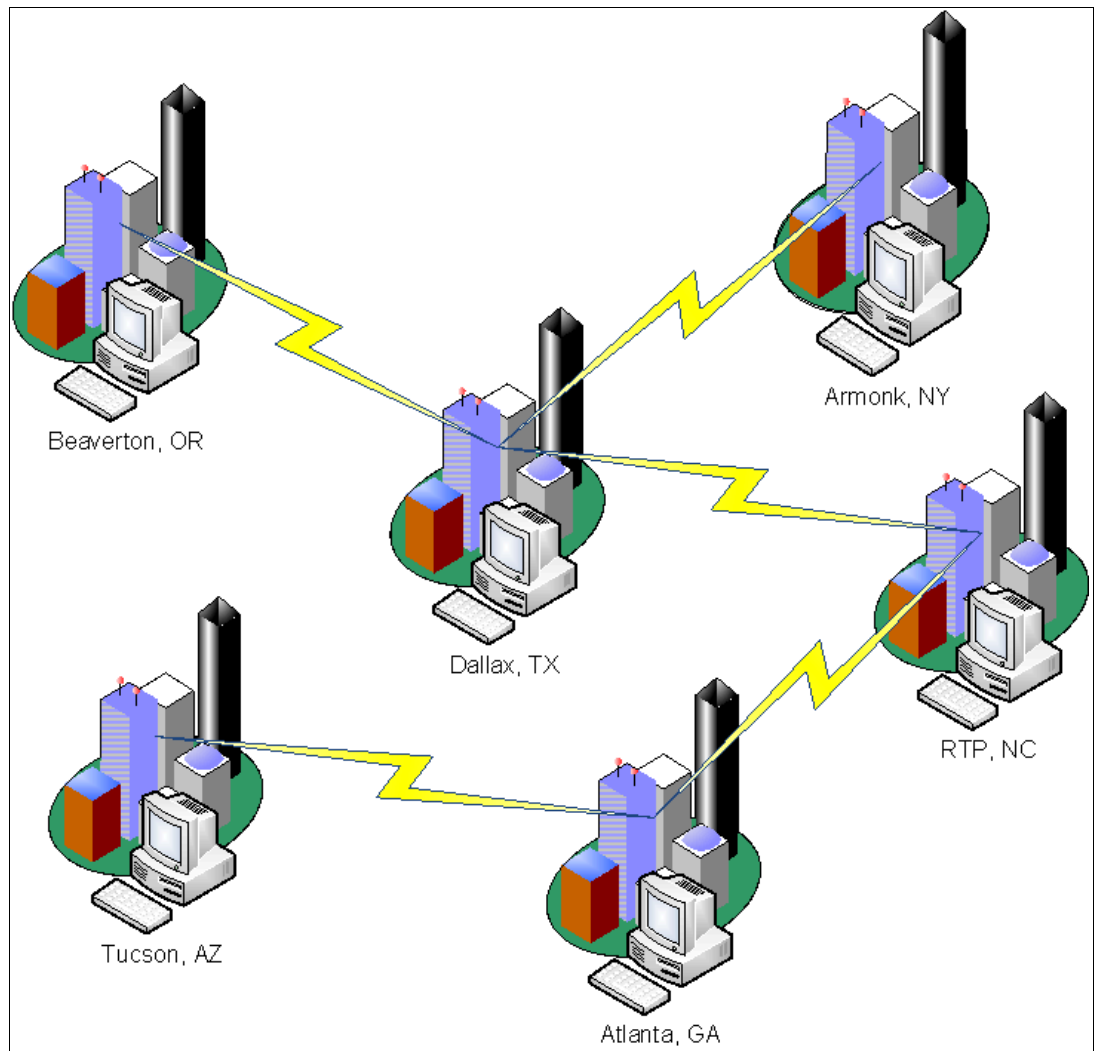


Figure 1-2 A WAN interconnecting LANs

1.3 Layer 1 networking concepts and terminology

Layer 1 of the OSI model is the layer at which the physical transmission of data occurs. The unit of transmission at Layer 1 is a bit. This section discusses some of the common concepts that are located at the Layer 1 level.

1.3.1 Ethernet cabling

Cabling in Ethernet is one of two forms, copper or fiber optic cabling. Copper is typically the least expensive choice for materials, components, and installation. Copper is commonly the media that is used to connect devices to the access layer switches.

Fiber optic cabling is more expensive than copper cabling. The optical components for devices and switches and the cost of any customer cabling is typically more expensive to install. However, the benefits of fiber optic cabling easily justifies the higher costs.

Fiber cabling provides for longer distance and is resistant to the signalling being distorted by electromagnetic interference.

Copper cabling

Ethernet utilizes two forms of copper cabling, coaxial and twisted pair.

Coaxial Cabling

Early Ethernet utilized coaxial copper cabling comes in two sizes:

- ▶ 10BASE-2 (RG58/RG6) Thinnet for distances up to 185 meters
- ▶ 10BASE-5 (RG8/11) Thicknet for distances up to 500 meters

For coaxial cabling, the common connector types that are used are BNC, as illustrated in Figure 1-3 or Type F, which is illustrated in Figure 1-4.



Figure 1-3 BNC Connector



Figure 1-4 Type F Connector

Twisted-Pair cabling

Twisted-pair copper cabling is a common media for Ethernet networking installations. Twisted-pair cabling is available as Unshielded Twisted-Pair (UTP) or Shielded Twisted-Pair (STP). This shielding helps to prevent electromagnetic interference.

There are also several categories of twisted-pair cabling. These categories indicate the signalling capabilities of the cabling, which we list in Table 1-1.

Table 1-1 TIA/EIA Cabling Categories

TIA/EIA cabling category	Maximum network speeds supported
Cat 1	Telephone or ISDN
Cat 2	4 Mb Token Ring
Cat 3	10 Mb Ethernet
Cat 4	16 Mb Token Ring
Cat 5	100 Mb Ethernet

TIA/EIA cabling category	Maximum network speeds supported
Cat 5e	1 Gb Ethernet
Cat 6	10 Gb Ethernet Short Distance - 55 m (180 ft)
Cat 6a	10 Gb Ethernet

The connector that is used for Ethernet twisted-pair cabling is likely the one most people recognize and associate with networking, the RJ45 connector, as illustrated in Figure 1-5.

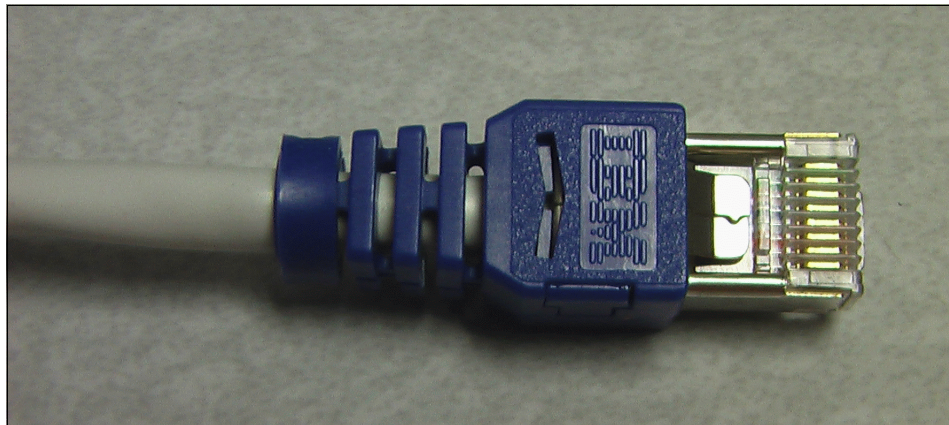


Figure 1-5 RJ45 copper connector

Twisted-pair cabling contains four pairs of wire inside of the cable, as shown in Figure 1-6.

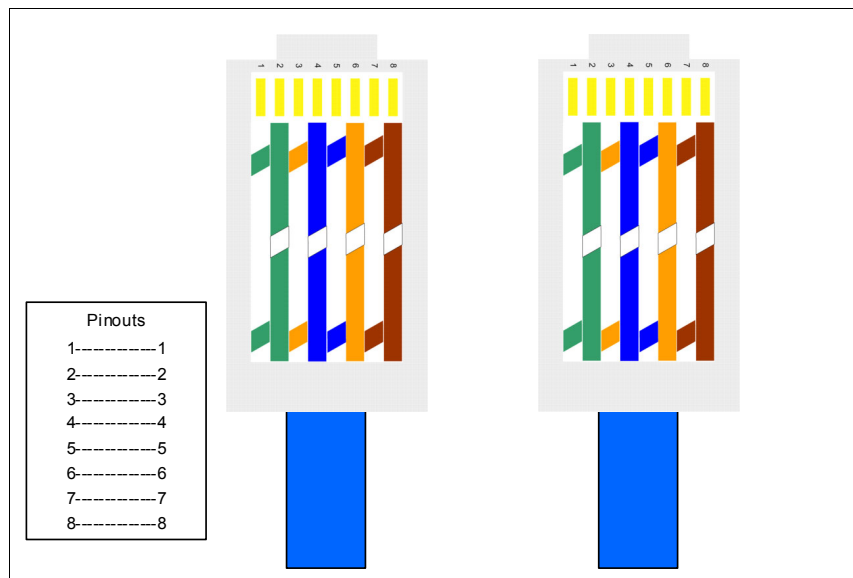


Figure 1-6 Straight through Ethernet cable

Ethernet operating in 10/100Mb mode only uses two pairs, pairs 1-2 and 3-6. Ethernet operating in 1Gb mode uses all four pairs, pairs 1-2, 3-6, 4-5, and 7-8. Distances up to 100 meters are supported.

Damaged twisted pairs: If a twisted-pair cable is damaged such that pair 4-5 or pair 7-8 cannot communicate, the link cannot communicate in 1 Gbps mode. If the devices are set to auto negotiate speed, the devices will successfully operate in 100 Mbps mode.

Distances of cabling segments: The actual maximum supported distances of a cabling segment vary on multiple factors, such as vendor support, cabling type, electromagnetic interference, and number of physical connections in the segment.

Twisted-pair crossover requirements

In 10/100 Mbps Ethernet operations, one pair of wire is used for data transmission and one pair is used for receiving data. When a device, such as a PC is attached to a hub or switch, the ports are designed so that the transmitting and receiving pairs are properly matched. When directly connecting two like devices, such as PC-to-PC, Hub-to-Hub, or Switch-to-Switch, a crossover in the pairs must be made.

Crossover function can be made internally by the port of one of the devices or can be achieved by using a crossover cable, as shown in Figure 1-7.

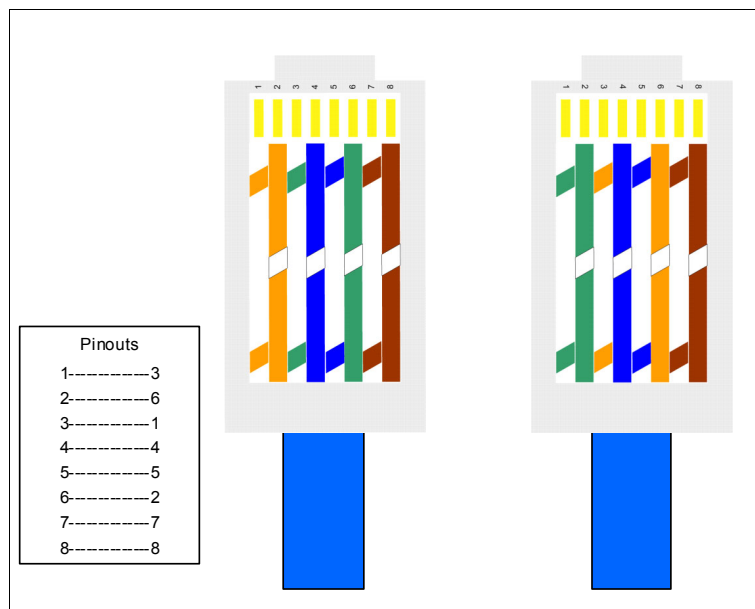


Figure 1-7 10/100 Mbps crossover cable

Ethernet ports without crossover are known as Medium Dependent Interface (MDI). Ports with crossover are known as Medium Dependent Interface Crossover (MDIX). The 'X' means crossover. Some ports have the ability to sense if crossover is needed and configures the port properly. This function is often referred to as Auto MDIX. For gigabit Ethernet, the auto crossover function is an optional part of the 1000 Base-T Ethernet standard.

Fiber Optic cabling

In copper cabling, electric signals are used to transmit data through the network. The copper cabling was the medium for that electrical transmission. In fiber optic cabling, light is used to transmit the data. Fiber optic cabling is the medium for channeling the light signals between devices in the network.

Two modes of optic signaling are discussed in this chapter, single mode and multi-mode. The main difference between the modes is the wavelength of the light used for the transmission as illustrated in Figure 1-8.

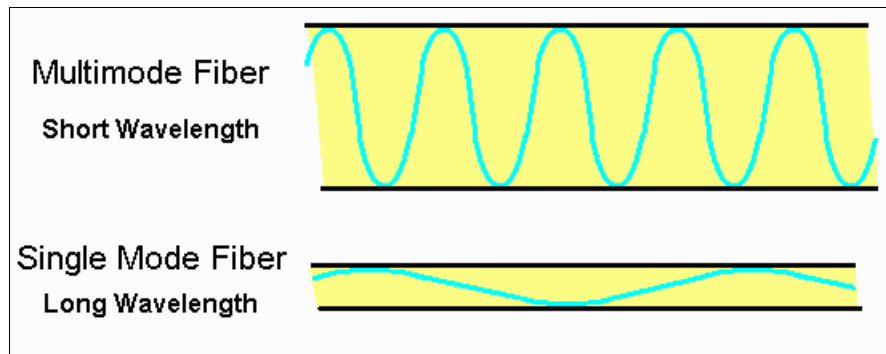


Figure 1-8 Multimode versus single-mode optic signaling

Single-mode fiber

Single-mode fiber (SMF) utilizes long wavelength light to transmit data and requires a cable with a small core for transmission as shown in Figure 1-8. The core diameter for single-mode cabling is nine microns in diameter as illustrated in Figure 1-9.

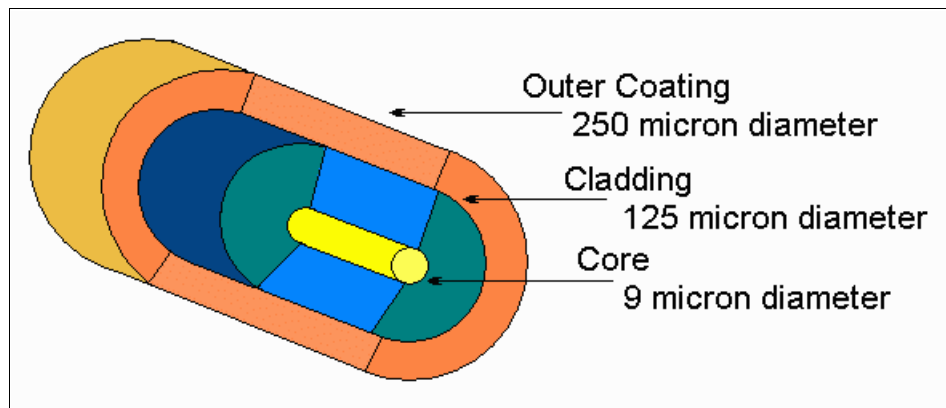


Figure 1-9 Single-mode fiber cable

Multimode Fiber

Multimode fiber (MMF) utilizes short wavelength light to transmit data and requires a cable with a larger core for transmission as shown in Figure 1-8. The core diameter for multimode cabling can be 50 or 62.5 microns in diameter. Refer to Figure 1-10 on page 9.

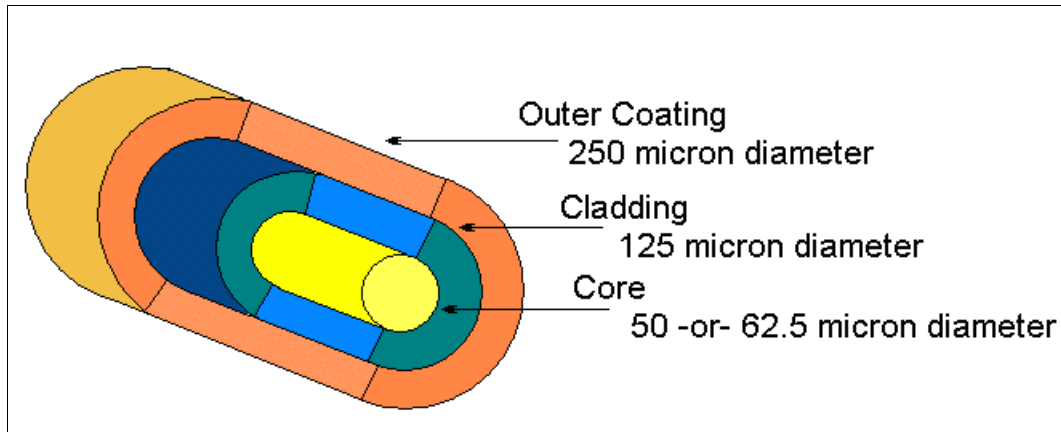


Figure 1-10 Multimode fiber cable

The color of the outer coating is sometimes used to identify if a cable is a multimode or single-mode, but this is not a reliable method. The TIA-598C standard suggests the outer coating to be yellow (Figure 1-12) for single-mode fiber and orange (Figure 1-13) for multimode fiber for civilian applications; however, this guideline is not always implemented, as shown in Figure 1-11, which shows a blue cable. The reliable method is to look on the outer coating of the cabling. The specifications of the cable are printed on the cable.



Figure 1-11 Blue 62.5 micron MMF cable

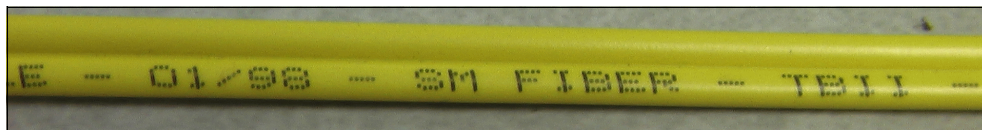


Figure 1-12 Yellow SMF cable



Figure 1-13 Orange 50 micron MMF cable

Connector types

The most common connector type for fiber-optic media used in networking today is the LC connector shown in Figure 1-14 on page 10.

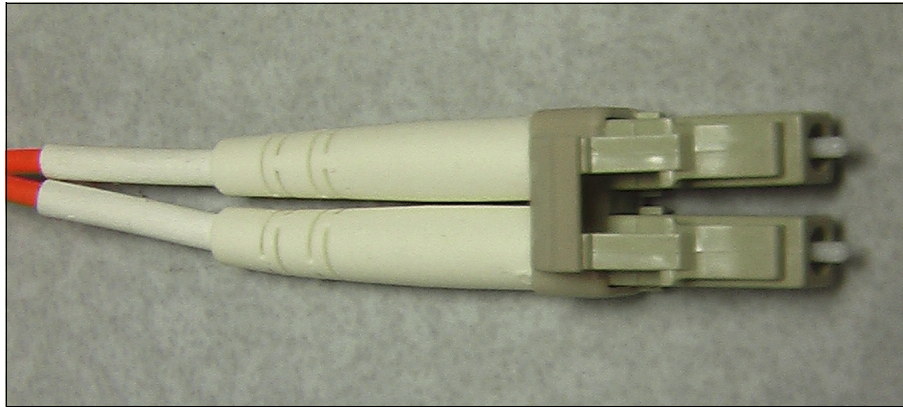


Figure 1-14 LC fiber connector

Other commonly encountered connectors in Ethernet networks are the SC connector (Figure 1-15) and the ST connector (Figure 1-16).

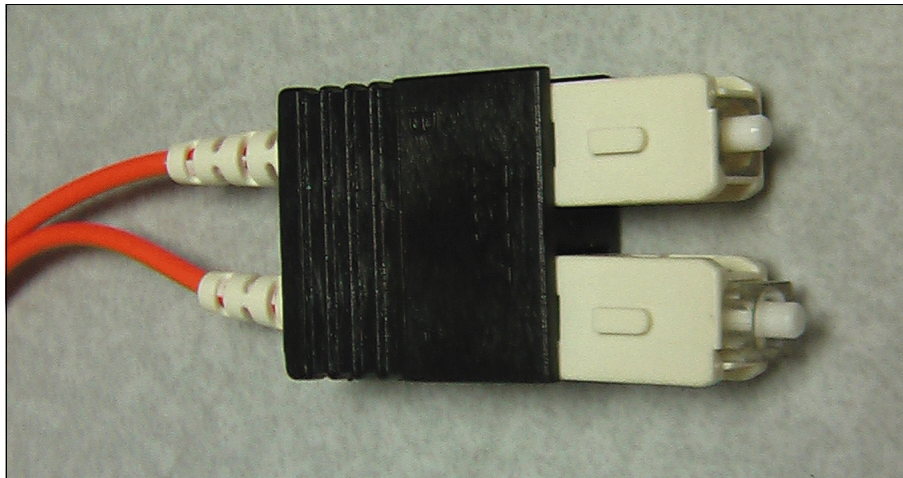


Figure 1-15 SC fiber connector



Figure 1-16 ST fiber connectors

Transceivers

A *transceiver* or *transmitter/receiver* is the fiber optic port of a device where the fiber optic cables connect. Occasionally, a device might have an integrated transceiver but this limits the flexibility in what type of cabling can be used. Most devices provide a slot for a modular

transceiver to be inserted, which provides flexibility for single or multimode implementations to be selected.

Some equipment can utilize a larger transceiver known as a *Gigabit Interface Converter* (GBIC), which Figure 1-17 illustrates. As technology advances, smaller transceivers are introduced that provide much higher port density, such as Small Form-factor Pluggable (SFP), 10 Gigabit SFP+, or 10 Gigabit Small Form-factor Pluggable (XFP). Figure 1-18 shows a comparison of the various transceivers.



Figure 1-17 Gigabit Interface Converter

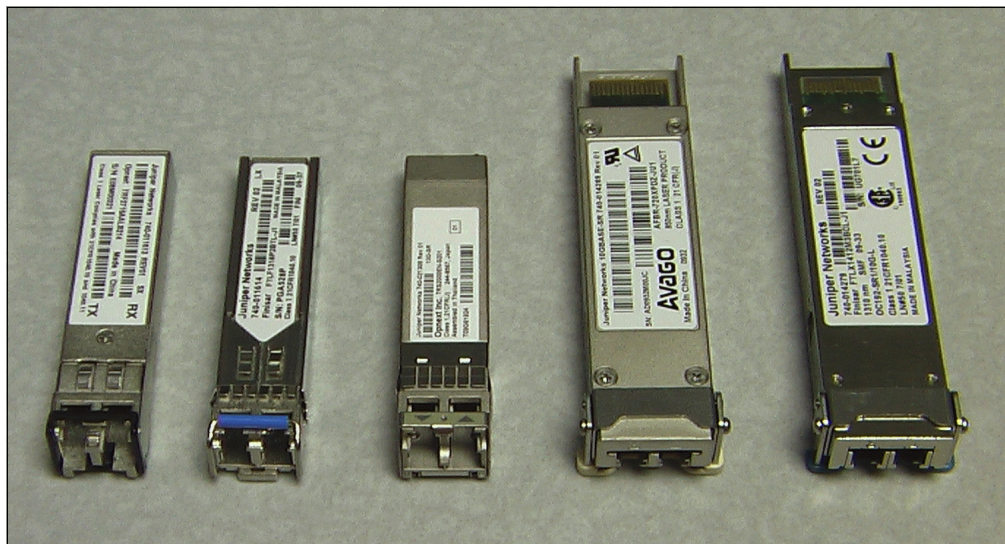


Figure 1-18 Left to right: SFP-MMF, SFP-SMF, SFP+-MMF, XFP-MMF, and XFP-SMF

1.3.2 Layer 1 components

The layer 1 components in Ethernet are devices that provide physical signal connectivity of the devices.

Hub

A network *hub* has the same functional properties of a segment using coax cabling to interconnect devices. It operates as a share media segment.

The difference is that the hub allows for much easier cabling. The cables plug into the hub, and the hub internally forms the shared media segment. A network hub connects multiple devices as a single shared media segment. All frames are viewable by all of the devices attached to the hub because it forms a shared media segment.

Repeater

A network *repeater* is similar in function to a network hub in that it forms a common shared media segment with all of the attached devices. The difference is a repeater receives the incoming signal and then “repeats” or regenerates the signal at new signal quality and strength as it is sent out to the other ports of the repeater.

In contrast, a hub is a passive device that allows the original signal to pass through and on to the other devices that are connected to the segment. A repeater is often used in environments where signal loss occurs in a segment because of media length or other environmental factors.

1.3.3 Network segments

A *network segment* can be thought of as the local layer 1 network to a device and any devices that are on the same common layer 1 network. The formal definition of a network segment is: A network segment is a portion of a network that is separated by a networking device, such as a router, switch, or bridge.

A network segment can be one of two types, a shared media or switched segment.

Shared media segment

Devices on a *shared media segment* must arbitrate for use of the media because multiple devices cannot simultaneously transmit. The devices must “share” the media; hence, the name shared media segment. Because all devices are connected to the same media (wire), all devices see communications from the other devices, as illustrated in Figure 1-19 on page 13. An increase in devices that are attached to the shared media segment means that there are more devices arbitrating for use of the segment to be able to transmit. Use of this type of segment does provide for a network that scales well.

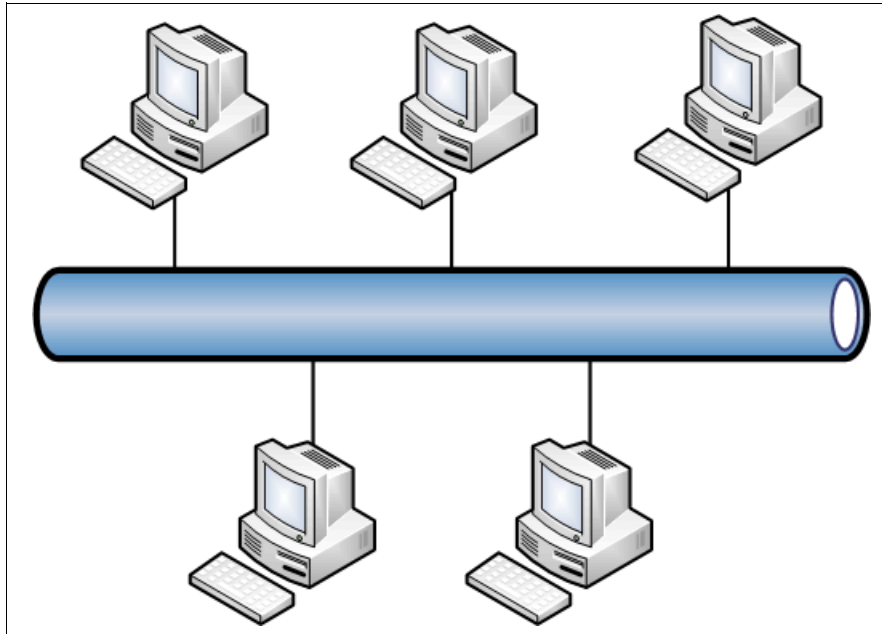


Figure 1-19 Shared media segment

The Carrier Sense Multiple Access / Collision Detection (CSMA/CD) access method is the protocol that Ethernet devices follow when operating in half duplex mode which is required for a shared media segment. There are three parts to the CSMA/CD Access Method.

Carrier Sense requires a device to “listen” on the media, wait until there are no other transmissions occurring, and then attempt to transmit. Multiple Access allows multiple nodes to be connected to the same media at the same time.

Collision Detection recognizes when two nodes attempt to transmit simultaneously and notifies the other nodes on the segment that the previous transmission is invalid. This notification function is known as *jamming*. Jamming consists of sending a special 32-bit sequence.

Switched segment

Devices on a *switched segment* can communicate directly to the network switch without any interference from other intermediate devices. Switched segments allow for full duplex communications.

1.3.4 Physical configuration parameters

When we discuss the physical layer (layer 1) properties, we consider elements, such as line speed and duplex.

Speed

Speed in Ethernet are rates, such as 10 Mbps, 100 Mbps, 1000 Mbps (1 Gbps), or 10 Gbps. Standards for higher speeds, such as 40 and 100 Gbps are in draft form now.

Duplex

Duplex modes are either full or half duplex. Half duplex is when a device can only send or receive at a given time (Figure 1-20 on page 14). Full duplex devices can send and receive

simultaneously (Figure 1-21). Also, when devices are connected to a shared media segment, all devices must operate in half duplex mode.

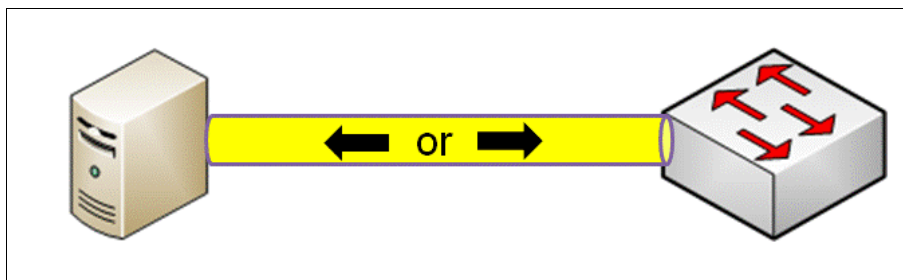


Figure 1-20 Half duplex

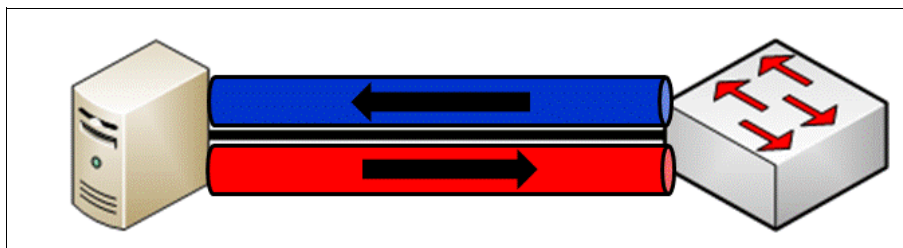


Figure 1-21 Full duplex

Autonegotiation

In Ethernet, the speed and duplex of the device that is attached to a segment must match. Autonegotiation of the speed and duplex of a device usually works well, but it is not 100% reliable. The problems usually occur with older 10/100 devices. Newer devices rarely have an issue negotiating with each other.

One key step to reduce negotiation problems is to make sure that both devices on a switch segment are configured the same. Either configure both devices for autonegotiation or “hard code” (manually configure) both the speed and duplex settings of both devices to the same settings.

1.3.5 Power over Ethernet (PoE)

Power over Ethernet (PoE) is the implementation of IEEE 802.3af, which allows both data and electric power to pass over a copper Ethernet LAN cable. This technology allows Voice over Internet Protocol (VoIP) telephones, wireless access points, video cameras, and point-of-sale devices to safely receive power from the same access ports that are used to connect personal computers to the network.

Power management mode

You can use the power management mode to determine the number of interfaces that can be provided with power. The following two factors constitute the power management mode:

Per port limit (PPL) The factor that decides the maximum power consumption that is permitted on a particular interface. If the power consumption by the powered device exceeds the specified value, PoE is shut down over that interface.

Power allocated for each interface

The factor that ensures that a certain amount of power is reserved for an individual interface from the total power budget for all interfaces. If at any point the total of the allocated power for all interfaces exceeds the total budget, the lower priority interfaces are turned off and the power allocated for those interfaces drops to 0.

There are two modes of power management:

Static

In this mode, the power that is allocated for each interface can be configured. The PPL value is the maximum value configured per interface.

Class

In this mode, the power allocation for interfaces is determined based on the class of powered devices that are connected. PPL is the maximum power value of the class of the powered device connected to the interface. The power allocated per interface is the maximum power of the powered device class, except for classes 0 and 3. For class 0 and class 3 powered devices, the momentary power consumption is considered as the power allocated for that interface. Therefore, PPL and power allocated per interface values change based on the powered device that is connected to the interface.

The default power management mode is Static mode.

Classes of powered devices

A powered device is classified based on the maximum power that it draws across all input voltages and operational modes. The most common class is 0, in which the switch allows a maximum draw of 15.4 W per port. The switch provides 15.4 W at the port to guarantee enough power to run a device, after accounting for line loss. Consider this example:

$$15.4 \text{ W} - \text{power loss (16\%)} = 12.95 \text{ W}$$

All 802.3af-compliant powered devices require no more than 12.95 watts.

Table 1-2 lists the classes of powered devices and associated power levels.

Table 1-2 PoE Classes

Class	Usage	Minimum power levels output from PoE port	Range of maximum power required by the powered device
0	Default	15.4 W	0.44 through 12.95 W
1	Optional	4.0 W	0.44 through 3.84 W
2	Optional	7.0 W	3.84 through 6.49 W
3	Optional	15.4 W	6.49 through 12.95 W

1.4 Layer 2 networking concepts and terminology

OSI Layer 2 or Data Link Layer provides the functional means for data transfer between adjacent nodes in the network. Layer 2 also provides the lowest level of addressability in an Ethernet network using Media Access Control (MAC) addresses. This section provides information about the fundamental concepts and terminology at Layer 2.

1.4.1 Frame structure

A *frame* is the unit of transmission at the layer 2 operations of a network. A frame is simply a series of bits. The groups of these bits are known as *fields* in the frame and have specific purposes (Figure 1-22).

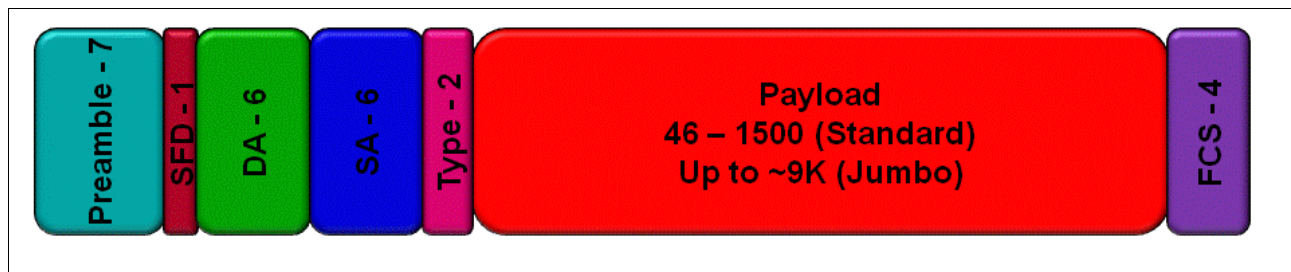


Figure 1-22 Ethernet Frame

A frame is composed of the following fields:

Preamble (7 bytes) Alternating sequence of 1 and 0 bit values used to synchronize the devices.

Start Frame Delimiter (SFD, 1 byte)
Byte containing 10101011, which signals the start of the frame that follows.

Destination Address (6 bytes)
MAC Address of the destination device for the frame. Three forms:

- Unicast: Unique address for individual device.
- Multicast: Address for a group of devices.
- Broadcast: Special multicast address destined for all devices.

Source Address (6 bytes)
MAC Address of the source device of the frame.

Ethernet Protocol Type or Length (2 bytes)
Use of this field is the distinction between 802.3 and Ethernet II (DIX). Values of x05DC (1500 decimal) or less indicates an 802.3 frame type and the value is the length of Payload field.
Values of x0600 and higher indicates an Ethernet II (DIX) frame type and the value is the Protocol Type.

Payload / Data (46 to 1500 bytes)
Data to be transferred from source to destination device. Normally up to 1500 bytes. If less than 46 bytes, then “padding” must be added to maintain minimum frame size.

Frame Check Sequence (FCS, 4 bytes)
Four byte cyclical redundancy check (CRC) value for error checking. Value is calculated DA through Payload fields by sending device. The

receiving device makes same calculation and compares value with that in the FCS field.

The maximum payload size for a standard Ethernet frame is 1500 bytes, and the total frame size of 1518 bytes.

Notice that this frame structure does not include any field to identify VLAN membership or frame priority. It is known as an “untagged” frame. For information about tagged frames, see “Tagged frames” on page 25.

1.4.2 Jumbo frames

A Jumbo frame is an expansion of the frame size up to approximately 9K. The maximum size support is vendor dependant. While some vendors even support beyond 9K, which is a common max size.

Jumbo frames allow for better performance by devices that do not need to process as many frames (frame headers). The result is less CPU utilization and less bandwidth used to transmit header information and less idle time of link due to interframe gaps.

Concurrent use of jumbo and standard frames sizes in some networks can impact performance of latency-sensitive applications. End-to-End network devices must support jumbo frames or else frames will be discarded.

1.4.3 Interframe gap

Ethernet devices must allow an idle period between frame transmissions, as illustrated in Figure 1-23. The minimum idle time is 96 bit times and is simply the amount of time required to send 96 bits.

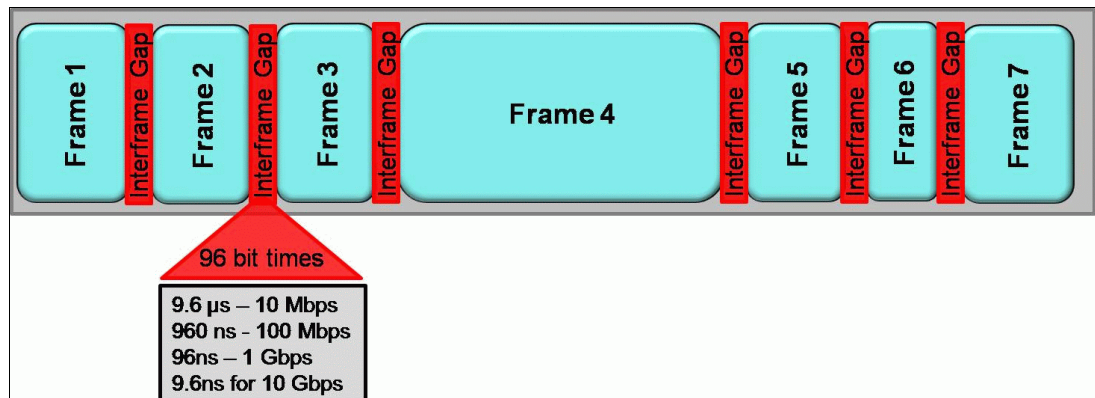


Figure 1-23 Interframe gap

The interframe gap (IFG) is also referred to as interframe spacing or interpacket gap (IPG).

1.4.4 Media Access Control addresses

Each frame contains a source and a destination Media Access Control (MAC) address. The device sending the frame includes its MAC as the source MAC and the destination MAC is the MAC address of the device the frame is targeted. The Network Interface Card (NIC) of each device communicating on the network has a MAC address that the manufacturer assigned.

Figure 1-24 shows several ways that a manufacturer might show a MAC address.

MAC Addresses

00:13:10:17:FB:F8 or 00-13-10-17-FB-F8
or 0013.1017.FBF8

- **First three (3) octets** are assigned by IEEE to NIC manufacturer as the OUI (Organizationally Unique Identifier).
- **Last three (3) octets** are assigned by manufacturer while maintaining uniqueness of address.

Figure 1-24 MAC addresses

The registered owner of an Organizationally Unique Identifier (OUI) can be searched at the following web site:

<http://standards.ieee.org/regauth/oui/index.shtml>

Within the first byte of a MAC address there are two bits that provide additional information, as illustrated in Figure 1-25 on page 19.

MAC Address Details

0013:10:17:FB:F8 or 0013-10-17-FB-F8
or 0013.1017.FBF8

The **first octet** in more detail:

b7	b6	b5	b4	b3	b2	b1	b0
----	----	----	----	----	----	----	----

- B0 – multicast bit:
 - 0 – unicast
 - 1 – multicast
- B1 – locally administered address bit:
 - 0 – globally unique (burned-in address)
 - 1 – locally administered

Figure 1-25 MAC address details

Looking at the first byte of a MAC address in more detail, there are specific functions of bit 0 and bit 1:

- ▶ Bit 0 of the first octet is the multicast bit. If this bit's value is 0, the frame is a unicast frame. If the value is 1, the frame is a multicast frame.
- ▶ Bit 1 of the first octet is the locally administered address bit. If this bit's value is 0, the MAC address is a globally unique address (manufacturer's assigned or burned-in address). If the value is 1, the MAC is a locally administered MAC assigned by the user/administrator.

Types of layer 2 traffic

There are three types of layer 2 traffic in Ethernet. The type is determined by the destination MAC address of the frame and are:

- ▶ Unicast (One to One)

Unicast is traffic with a specific device's MAC address as the destination MAC in the frame. The address can be either the burned-in or a locally administered MAC address. The device with the same MAC address as the destination processes the incoming frame.

- ▶ Multicast (One to Many)

Multicast is traffic with a multicast MAC address as the destination MAC in the frame. The multicast MAC address is sometimes referred to as a group MAC address. The actual destination devices are any devices configured to process frames with that specific group MAC address. Some examples are:

- 01-80-C2-00-00-00 used by Spanning Tree Protocol (IEEE 802.1d)
- 01-00-5E-xx-xx-xx used by IPv4 Multicast Addresses

- Broadcast (One to Every)

Broadcast is traffic with a special case of multicast frame in which the group address is a special address, FF-FF-FF-FF-FF-FF, as the destination MAC in the frame. All layer 2 devices on the network must process the broadcast frame.

1.4.5 Bridging or Layer 2 switching

A bridge connects multiple segments at the Data Link Layer (OSI Layer 2), as shown in Figure 1-26.

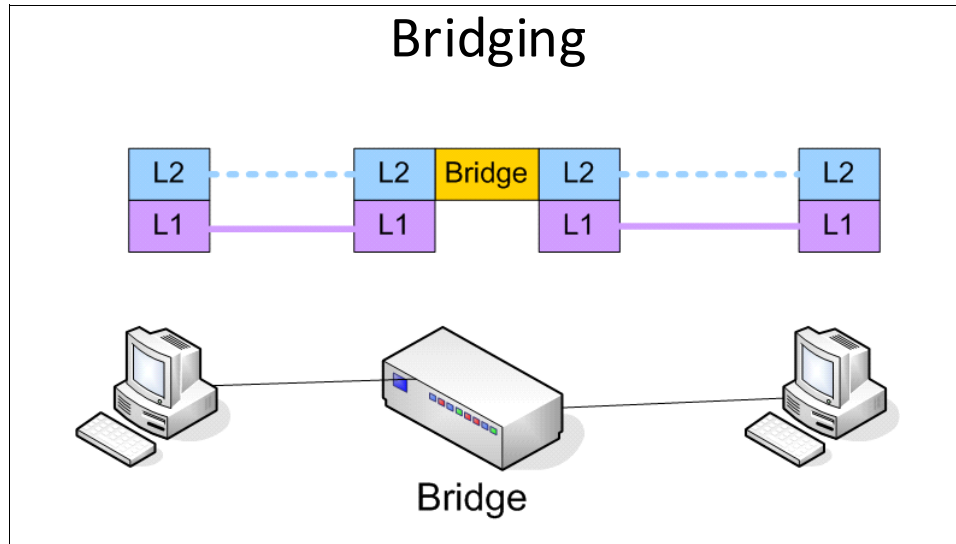


Figure 1-26 Bridge or Layer 2 switch

Another way to express the same idea is that a bridge maintains one broadcast domain (Layer 2) while isolating collision domains (Layer 1). Bridges began as PCs with multiple NICs installed with a special bridging application running that performs the bridging function. MAC addresses of attached devices are learned on each port.

What is the difference between a bridge and a layer 2 switch? From a functional standpoint, the answer is that *there is no difference*. A simple way to explain the difference is that a bridge is a device that performs the function of maintaining the broadcast domain while isolating the collision domain.

A Layer 2 switch is a device that implements the bridging function primarily in hardware. Layer 2 switches have most of the bridging function incorporated into the chipsets that forward the frames. This typically provides for wire speed forwarding rates of traffic. The speed and port density of a switch is much more than the old bridge implementations on PCs.

The MAC addresses of devices attached to a switch are added to the MAC table of the switch along with the port on which the MAC address was learned. This method is accomplished by monitoring incoming source MAC addresses and adding any new MAC addresses to the table. If a device moves from one port to another, the MAC entry in the table is updated with the new port. Also, if a device is powered off, removed from the network, or for some other reason is not sending traffic on the network, the MAC entry is removed. The removal occurs after the time at which any traffic from this MAC address was received or exceeded an expiration time value of the switch.

Frame forwarding

In this section, we discuss how a Layer 2 switch forwards traffic in various cases.

Multicast or broadcast frames

The first case, illustrated in Figure 1-27, describes the process for forwarding a multicast or broadcast frame that is sent from a device with MAC A attached to Port 1 of the switch. Upon receiving the frame, the switch copies the frame and sends it out to all other ports.

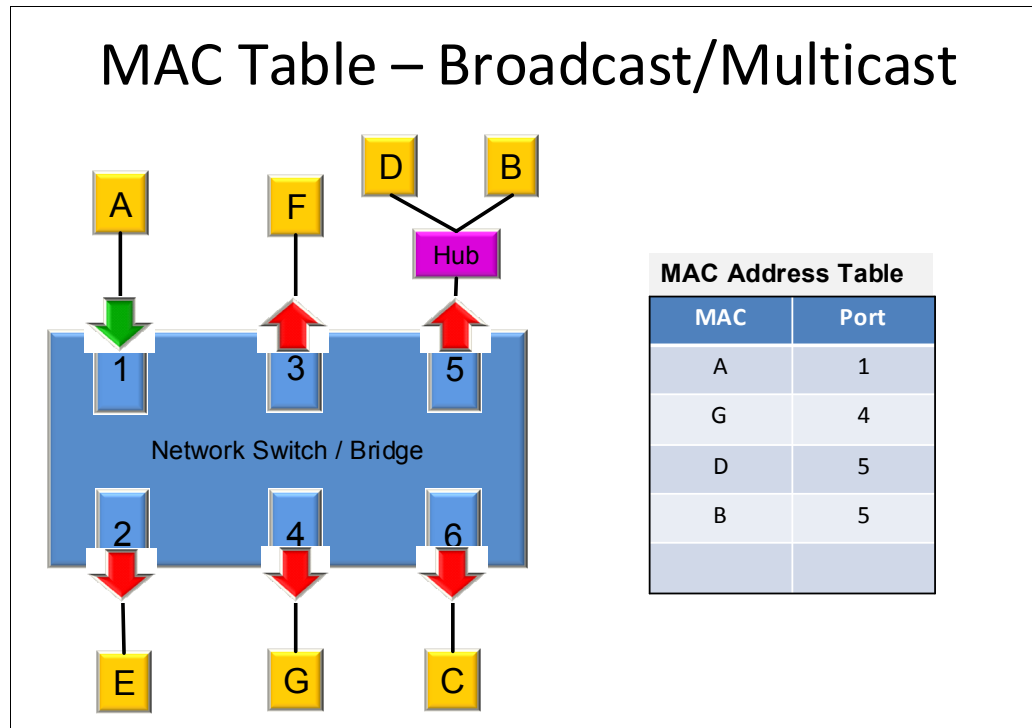


Figure 1-27 Broadcast/multicast frame forwarding

Unicast frame: MAC learned

The next case, illustrated in Figure 1-28 on page 22, describes the process for forwarding a unicast frame with the destination MAC G received from the device on Port 1 of the switch. Upon receiving the frame, the switch queries the MAC address table in the switch for the destination MAC address. If the MAC is found in the table, the frame is forwarded out the port associated with the MAC in the table. In this example, the frame is forwarded to port 4 because the MAC table has an entry for MAC G on port 4.

The MAC address entries are dynamically populated by monitoring the source MAC address of frames coming into the ports. MAC entries age out of the table; therefore, devices that are not sending frames are removed. The age-out time is vendor dependent and some allow for the time out value to be a configurable parameter. The age-out time also varies depending on other protocols or features enabled on the switch.

If a device is moved from one port to another port, the MAC is learned and the old entry in the MAC table is deleted, and a new entry is added with the new port.

Unicast Forwarding (MAC Learned)

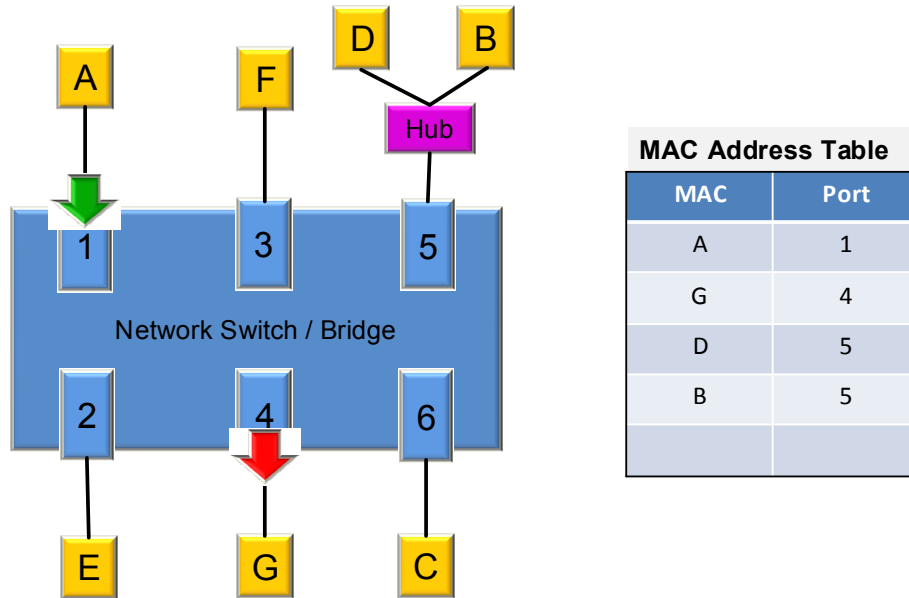


Figure 1-28 Unicast frame forwarding: MAC learned

Unicast frame: MAC unknown

The next case, illustrated in Figure 1-29 on page 23, describes the process for forwarding an unicast frame sent from a device with destination MAC C (not in MAC Address Table of switch) sent from a device with MAC A on Port 1 of the switch. Upon receiving the frame, the switch queries the MAC address table in the switch for the destination MAC address. If the MAC is *not* found in the table, the frame is copied and forwarded out to all other ports. In this example, the frame is forwarded out to all other ports of the switch.

This situation is not the same as a broadcast because the frames that are forwarded are unicast frames (frame with the specific MAC address of the destination device). Some vendors refer to the process as *flooding*. The frames are sent down all paths to reach the destination device.

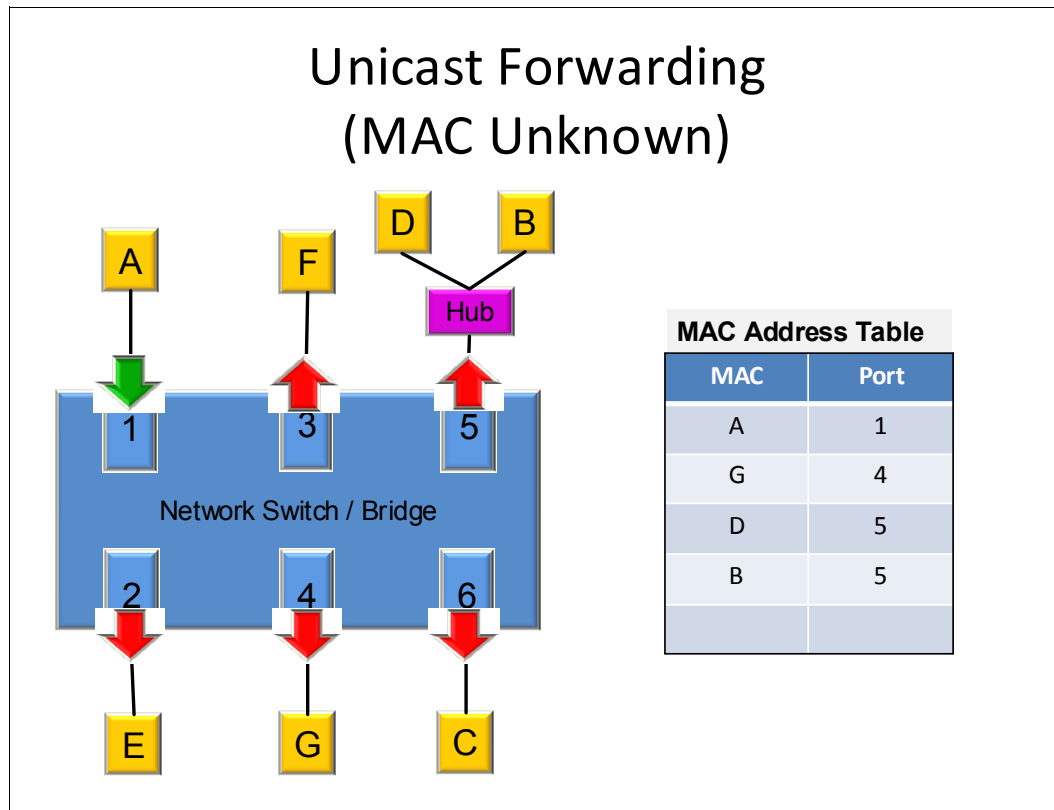


Figure 1-29 Unicast frame forwarding: MAC unknown

Internet Group Management Protocol

Internet Group Management Protocol (IGMP) is a protocol that provides management of multicast traffic in a Layer 2 network. The protocol allows multicast traffic for a specific multicast group to only be sent out ports of switches that have clients using the groups multicast traffic.

1.4.6 Virtual Local Area Network

A Virtual Local Area Network (VLAN) is a networking concept in which a network is logically divided into smaller virtual LANs. The Layer 2 traffic in one VLAN is logically isolated from other VLANs and is illustrated in Figure 1-30 on page 24.

VLANs - Isolation at Layer 2

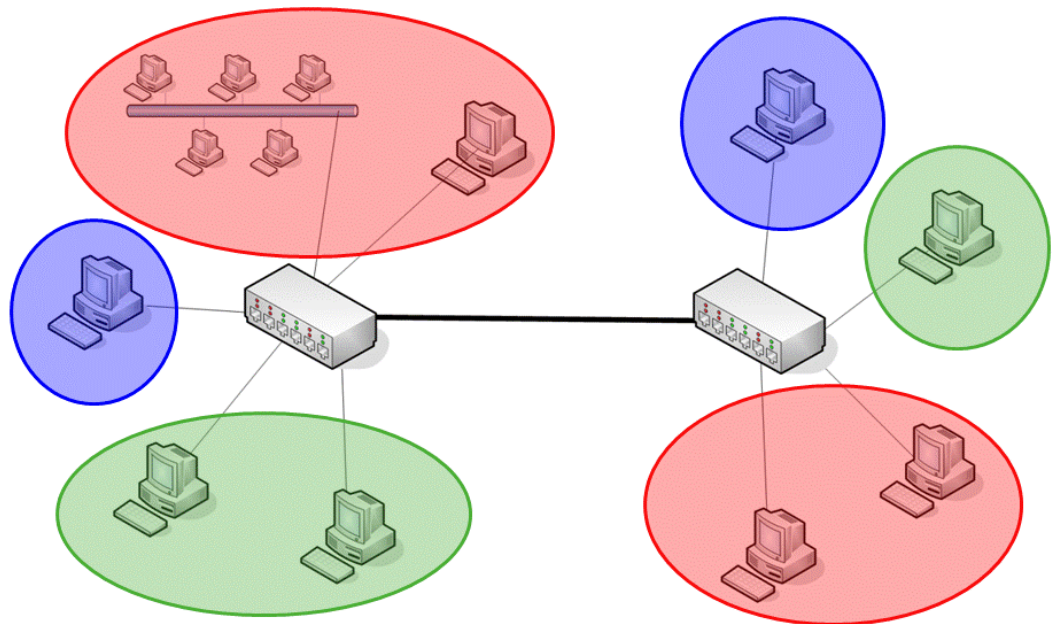


Figure 1-30 VLANs: Isolation at Layer 2

There are two methods for maintaining isolation of VLAN traffic between switches, as shown in Figure 1-31.

VLAN tagging

Logical isolation of the VLAN traffic between devices can occur through two methods.

One link for each VLAN:

- NOT scalable.
- Inefficient use of link capacity

VLAN tagging:

- Scalable
- Efficient use of link capacity

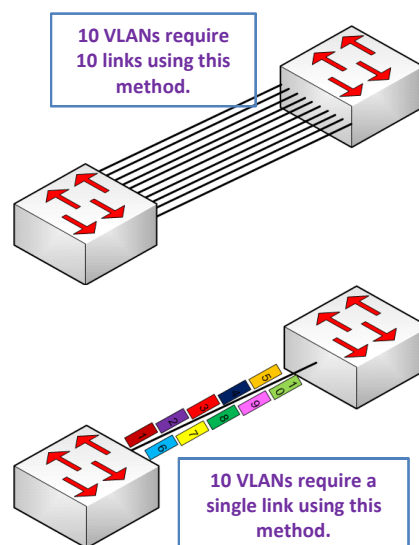


Figure 1-31 VLAN tagging

The first method for maintaining isolation of VLAN traffic between switches is using a single link for each VLAN. This method does not scale well because it consumes many ports in networks with multiple VLANs and multiple switches. This method also does not utilize link capacity in an efficient manner when traffic in the VLANs are not uniform.

The second method is VLAN tagging over a single link where each frame is tagged with its VLAN ID. This method is highly scalable because only a single link is required to provide connectivity to many VLANs, which provides a for better utilization of the link capacity when VLAN traffic is not uniform.

The protocol for VLAN tagging of frames in a LAN environment is defined by the IEEE 802.1 P/Q standard.

Inter-Switch Link: Inter-Switch Link (ISL) is another protocol for providing VLAN tagging function in a network. This protocol is not compatible with the IEEE 802.1P/Q standard.

Tagged frames

The IEEE 802.1P/Q standard provides a methodology for added information, such as VLAN membership and priority to the frame, as shown in Figure 1-32.

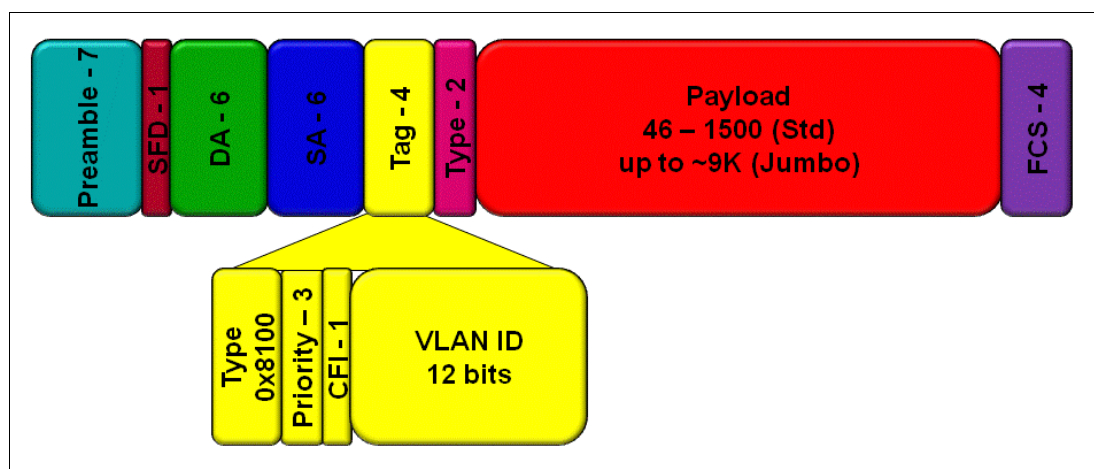


Figure 1-32 IEEE 802.1 P/Q tagged Ethernet frame

The standard provides an additional 4 bytes of information to be added to each Ethernet frame. A frame including this extra information is known as a *tagged frame*.

The 4 byte tag has four component fields:

- ▶ A type field that is 2 bytes long with the hexadecimal value of x8100 to identify the frame as an 802.1P/Q tagged frame
- ▶ A priority field that is 3 bits long to allow a priority value of eight different values to be included in the tag. This is the “P” portion of the 802.1P/Q standard
- ▶ A Canonical Format Indicator field that is 1 bit long to identify when the contents of the payload field are in canonical format
- ▶ A VLAN ID field that is 12 bits long to identify which VLAN the frame is a member of, with 4096 different VLANs possibilities

1.4.7 Interface VLAN operation modes

Interfaces on a switch can operate in two VLAN modes, single VLAN mode or multiple VLAN mode.

Single VLAN mode

Single VLAN mode operation is also referred to as *access mode*. A port operating in this mode is associated with a single VLAN. Incoming traffic does not have any VLAN identification. Because the untagged frames enter the port, the VLAN identification for the VLAN that is configured for the port is added to the inbound frames.

Switch ports: Some vendors use terms other than access mode for ports operating in single VLAN mode. Those vendor's switch ports might be configured to operate in single VLAN mode by configuring a Port VLAN ID (PVID) and adding the port as a member of the VLAN.

Multiple VLANs mode

Multiple VLAN mode operation is also referred to as *trunk mode*. A port operating in this mode can receive frames that have VLAN tags. The port is also configured with the VLANs to which the port is allowed to send and receive frames.

The IEEE 802.1Q specification allows untagged traffic on a multi-VLAN port to be associated with a single VLAN, which is referred to as the native VLAN for the port, as shown in Figure 1-33.

This provision allows traffic with no VLAN tag to be received and associated with the VLAN that is configured as the PVID or native VLAN. Outbound traffic for this VLAN on a port configured in this manner is transmitted with no tag so that the receiving device can receive the frame in an untagged format. This method provides compatibility with existing devices or devices that are configured in single VLAN mode and attached to a port that is configured as a multi-VLAN port.

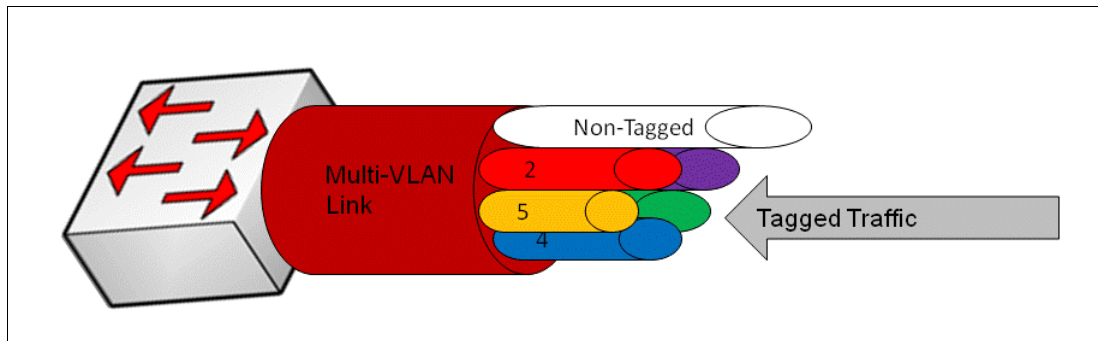


Figure 1-33 Multiple VLAN mode link

Varied meanings of the word trunk: The term trunk is used to express separate ideas in the networking industry. When using this term understand that others might use the term in a different manner. The term trunk can mean a port operating in multiple VLAN mode, or it can mean link aggregated port.

1.4.8 Link aggregation

Link aggregation combines multiple physical links to operate as a single larger logical link. The member links no longer function as an independent physical connections but as a members of the larger logical link. See Figure 1-34.

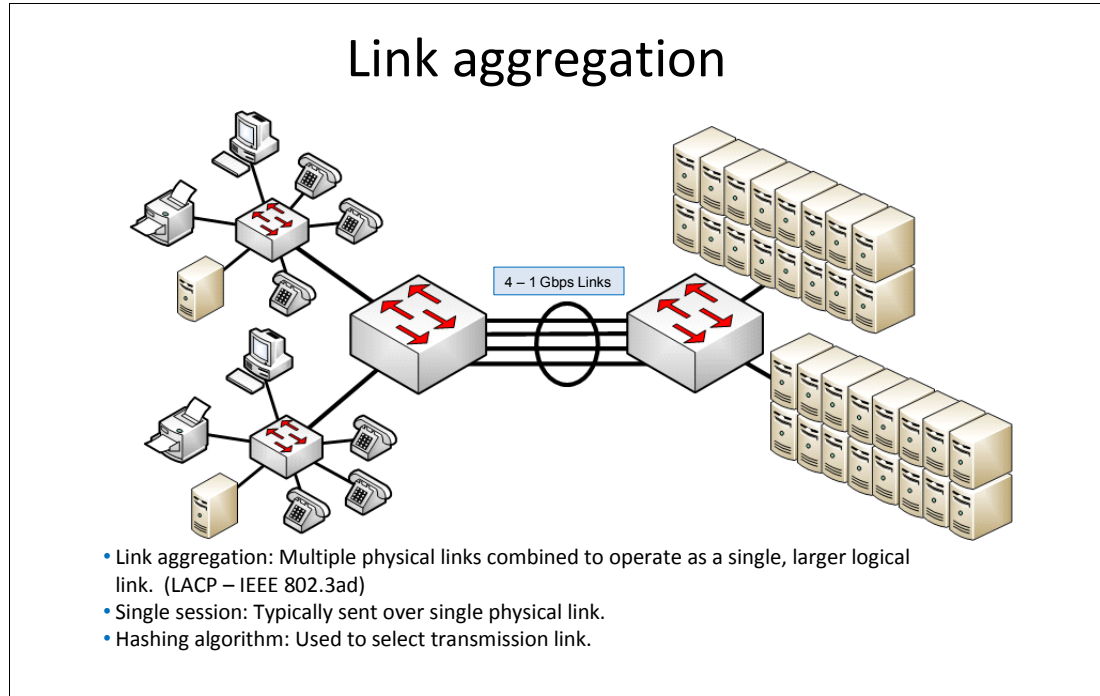


Figure 1-34 Link aggregation

Link aggregation provides greater bandwidth between the devices at each end of the aggregated link. Another advantage of link aggregation is increased availability because the aggregated link is composed of multiple member links. If one member link fails, the aggregated link continues to carry traffic over the remaining member links.

Each device that is interconnected by the aggregated link uses a hashing algorithm to determine on which of the member links frames can be transmitted. The hashing algorithm can use varying information in the frame to make the decision, which might include source MAC, destination MAC, source IP, destination IP, or a combination of these values.

1.4.9 Spanning Tree Protocol

Spanning Tree Protocol (STP) provides Layer 2 loop prevention and is commonly found in various forms, such as existing Spanning Tree Protocol (STP), Rapid Spanning Tree Protocol (RSTP), Multiple Spanning Tree Protocol (MSTP), and VLAN Spanning Tree Protocol (VSTP). A common default spanning-tree protocol is RSTP, which provides faster convergence times than STP. However, some existing networks require the slower convergence times of basic STP.

The operation of STP

STP uses bridge protocol data unit (BPDU) packets to exchange information with other switches. BPDUs send hello packets out at regular intervals to exchange information across bridges and to detect loops in a network topology.

There are two types of BPDUs:

- ▶ **Configuration BPDUs:** Contain configuration information about the transmitting switch and its ports, including switch and port MAC addresses, switch priority, port priority, and port cost.
- ▶ **Topology Change Notification (TCN) BPDUs:** When a bridge needs to signal a topology change, it starts to send TCNs on its root port. The designated bridge receives the TCN, acknowledges it, and generates another one for its own root port. The process continues until the TCN reaches the root bridge.

STP uses the information provided by the BPDUs to elect a root bridge, identify root ports for each switch, identify designated ports for each physical LAN segment, and prune specific redundant links to create a loop-free tree topology. All leaf devices calculate the best path to the root device and place their ports in blocking or forwarding states based on the best path to the root. The resulting tree topology provides a single active Layer 2 data path between any two end stations.

Rapid Spanning Tree Protocol

RSTP provides better reconvergence time than the original STP. RSTP identifies certain links as point-to-point. When a point-to-point link fails, the alternate link can transition to the forwarding state.

An RSTP domain has the following components:

Root port	The “best path” to the root device.
Designated port	Indicates that the switch is the designated bridge for the other switch connecting to this port.
Alternate port	Provides an alternate root port.
Backup port	Provides an alternate designated port.

RSTP was originally defined in the IEEE 802.1w draft specification and later incorporated into the IEEE 802.1D-2004 specification.

Multiple Spanning Tree Protocol

Although RSTP provides faster convergence time than STP, it still does not solve a problem inherent in STP, that all VLANs within a LAN must share the same spanning tree. To solve this problem, we use MSTP to create a loop-free topology in networks with multiple spanning-tree regions.

An MSTP region allows a group of bridges to be modeled as a single bridge. An MSTP region contains multiple spanning tree instances (MSTIs). MSTIs provide different paths for different VLANs. This functionality facilitates better load sharing across redundant links.

An MSTP region can support up to 64 MST instances, and each instance can support anywhere from 1 through 4094 vlans.

MSTP was originally defined in the IEEE 802.1s draft specification and later incorporated into the IEEE 802.1Q-2003 specification.

VLAN Spanning Tree Protocol

With VSTP, switches can run one or more STP or RSTP instances for each VLAN on which VSTP is enabled. For networks with multiple VLANs, VSTP enables more intelligent tree spanning because each VLAN can have interfaces enabled or disabled depending on the paths that are available to that specific VLAN.

By default, VSTP runs RSTP, but you cannot have both standalone RSTP and VSTP running simultaneously on a switch. VSTP can be enabled for up to 253 VLANs.

BPDU protection

BPDU protection can help prevent STP misconfigurations that can lead to network outages. Receipt of BPDUs on certain interfaces in an STP, RSTP, VSTP, or MSTP topology, can lead to network outages.

BPDU protection is enabled on switch interfaces connected to user devices or on interfaces on which no BPDUs are expected, such as edge ports. If BPDUs are received on a protected interface, the interface is disabled and stops forwarding frames.

Loop protection

Loop protection increases the efficiency of STP, RSTP, VSTP, and MSTP by preventing ports from moving into a forwarding state that might result in a loop opening up in the network.

A blocking interface can transition to a forwarding state in error if the interface stops receiving BPDUs from its designated port on the segment. Such a transition error can occur when there is a hardware error on the switch or software configuration error between the switch and its neighbor.

When loop protection is enabled, the spanning-tree topology detects root ports and blocked ports and makes sure that both keep receiving BPDUs. If a loop-protection-enabled interface stops receiving BPDUs from its designated port, it reacts as it might react to a problem with the physical connection on this interface. It does not transition the interface to a forwarding state, but instead transitions it to a loop-inconsistent state. The interface recovers and then it transitions back to the spanning-tree blocking state as soon as it receives a BPDU.

We recommend that you enable loop protection on all switch interfaces that have a chance of becoming root or designated ports. Loop protection is most effective when enabled in the entire switched network. When you enable loop protection, you must configure at least one action (**alarm**, **block**, or both).

An interface can be configured for either loop protection or root protection, but not for both.

Root protection

Root protection increases the stability and security of STP, RSTP, VSTP, and MSTP by limiting the ports that can be elected as root ports. A root port elected through the regular process has the possibility of being wrongly elected. A user bridge application running on a PC can generate BPDUs too and interfere with root port election. Using root protection, network administrators can manually enforce the root bridge placement in the network.

Root protection is enabled on interfaces that must not receive superior BPDUs from the root bridge and must not be elected as the root port. These interfaces become designated ports and are typically located on an administrative boundary. If the bridge receives superior STP BPDUs on a port that has root protection enabled, that port transitions to a root-prevented STP state (inconsistency state), and the interface is blocked. This blocking prevents a bridge that must not be the root bridge from being elected the root bridge. After the bridge stops receiving superior STP BPDUs on the interface with root protection, the interface returns to a listening state, followed by a learning state, and ultimately back to a forwarding state. Recovery back to the forwarding state is automatic.

When root protection is enabled on an interface, it is enabled for all of the STP instances on that interface. The interface is blocked only for instances for which it receives superior BPDUs. Otherwise, it participates in the spanning-tree topology.

An interface can be configured for either root protection or loop protection, but not for both.

1.4.10 Link Layer Discovery Protocol

Link Layer Discovery Protocol (LLDP) is a vendor-independent protocol for network devices to advertise information about their identity and capabilities. It is referred to as *Station and Media Access Control Connectivity Discovery*, which is specified in the 802.1ab standard. With LLDP and Link Layer Discovery Protocol–Media Endpoint Discovery (LLDP-MED), network devices can learn and distribute device information on network links. With this information, switches can quickly identify a variety of devices, resulting in a LAN that interoperates smoothly and efficiently.

LLDP-capable devices transmit information in Type Length Value (TLV) messages to neighbor devices. Device information can include specifics, such as chassis and port identification and system name and system capabilities. The TLVs use this information from parameters that were already configured in the Junos software from Juniper Networks.

LLDP-MED goes one step further, exchanging IP-telephony messages between the switch and the IP telephone. These TLV messages provide detailed information about PoE policy. The PoE Management TLVs let the switch ports advertise the power level and power priority needed, for example, the switch can compare the power needed by an IP telephone running on a PoE interface with available resources. If the switch cannot meet the resources required by the IP telephone, the switch can negotiate with the telephone until a compromise on power is reached.

The switch also uses these protocols to ensure that voice traffic gets tagged and prioritized with the correct values at the source itself, for example, 802.1p CoS and 802.1Q tag information can be sent to the IP telephone.

1.4.11 LLDP TLVs

Basic TLVs are:

Chassis Identifier	The MAC address associated with the local system.
Port identifier	The port identification for the specified port in the local system.
Port Description	The user-configured port description. The port description can be a maximum of 256 characters.
System Name	The user-configured name of the local system. The system name can be a maximum of 256 characters.
System Description	The system description containing information about the software and current image running on the system. This information is not configurable but taken from the software.
System Capabilities	The primary function performed by the system. The capabilities that the system supports, for example, bridge or router. This information is not configurable but based on the model of the product.
Management Address	The IP management address of the local system.

Additional 802.3 TLVs include:

Power via MDI A TLV that advertises MDI power support, PSE power pair, and power class information.

MAC/PHY Configuration Status

A TLV that advertises information about the physical interface, such as autonegotiation status and support and MAU type. The information is not configurable but based on the physical interface structure.

Link Aggregation A TLV that advertises if the port is aggregated and its aggregated port ID.

Maximum Frame Size

A TLV that advertises the Maximum Transmission Unit (MTU) of the interface sending LLDP frames.

Port VLAN A TLV that advertises the VLAN name configured on the interface.

LLDP-MED provides the following TLVs:

LLDP MED Capabilities

A TLV that advertises the primary function of the port. The capabilities values range 0 through 15:

0	Capabilities
1	Network Policy
2	Location Identification
3	Extended Power via MDI-PSE
4	Inventory
5 through 15	Reserved

LLDP-MED Device Class Values

0	Class not defined
1	Class 1 Device
2	Class 2 Device
3	Class 3 Device
4	Network Connectivity Device
5 through 255	Reserved

Network Policy A TLV that advertises the port VLAN configuration and associated Layer 2 and Layer 3 attributes. Attributes include the policy identifier, application types, such as voice or streaming video, 802.1Q VLAN tagging, and 802.1p priority bits and Diffserv code points.

Endpoint Location A TLV that advertises the physical location of the endpoint.

Extended Power via MDI

A TLV that advertises the power type, power source, power priority, and power value of the port. It is the responsibility of the PSE device (network connectivity device) to advertise the power priority on a port.

1.5 Layer 3 networking concepts and terminology

At the Layer 2 level of the network, the function was to provide a means of communication to adjacent nodes in the network or node-to-node communications. Layer 3 functions of the Ethernet network provides the means for communications from the source to the destination. These functions includes path determination or routing between Layer 3 logical networks. For TCP/IP networks, the logical Layer 3 network is also known as subnets. A common network design is to have a single IP subnet configured for each VLAN, as illustrated in Figure 1-35.

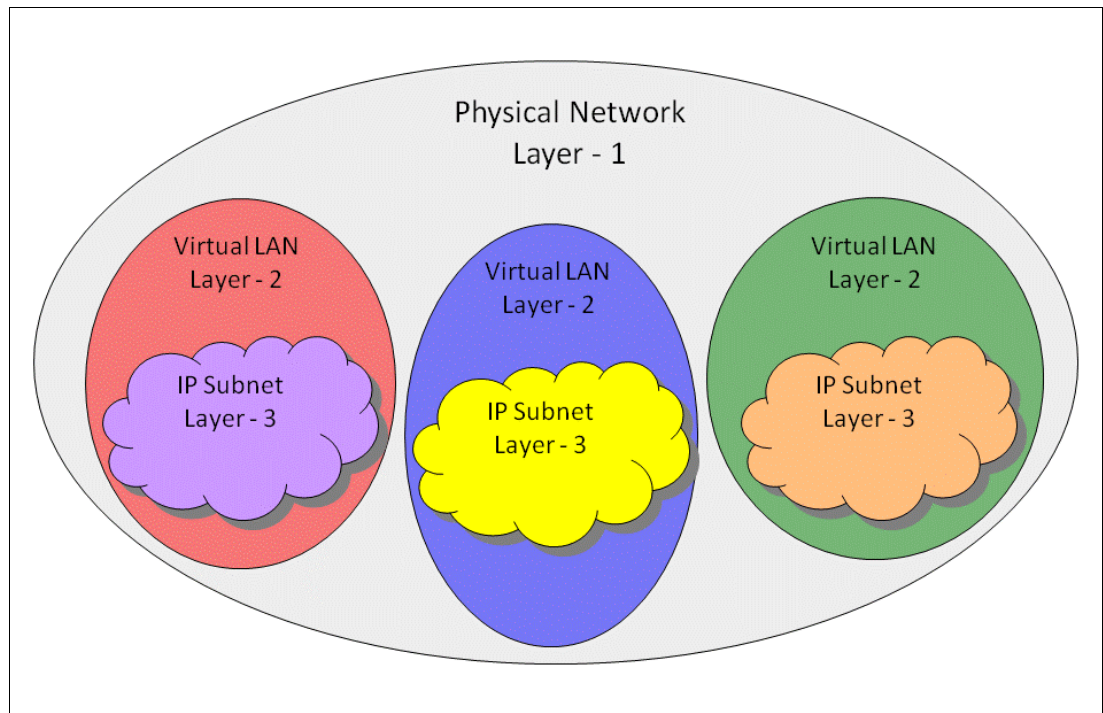


Figure 1-35 Single IP subnet per VLAN

Another design option is to have multiple IP subnets configured on a single VLAN, which is illustrated in Figure 1-36 on page 33.

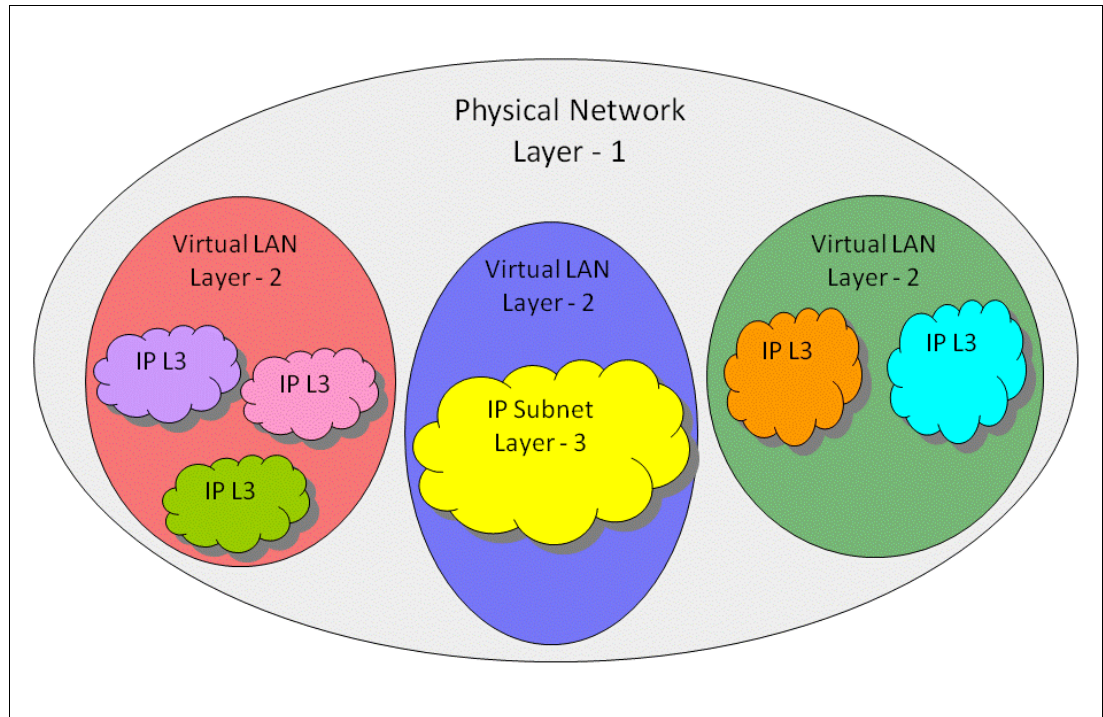


Figure 1-36 Multiple IP subnets per VLAN

1.5.1 IP routing

A router forwards Layer 3 packets from one logical network (IP subnet) to another, as shown in Figure 1-37. A router provides for connectivity at the Layer 3 level of the network while broadcasts are isolated to the separated Layer 2 networks or broadcast domains.

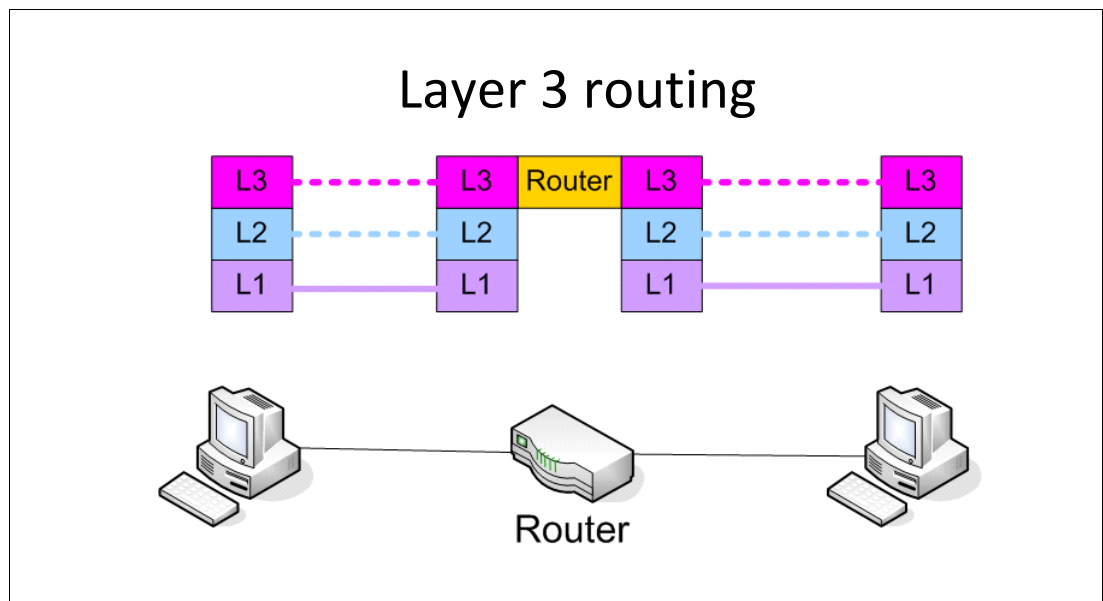


Figure 1-37 Layer 3 routing

The IP routing or IP forwarding process can have a number of steps, depending on the features and security functions that are enabled. The core process followed is shown in

Figure 1-38, which does not include every step but shows a high-level view of the forwarding process.

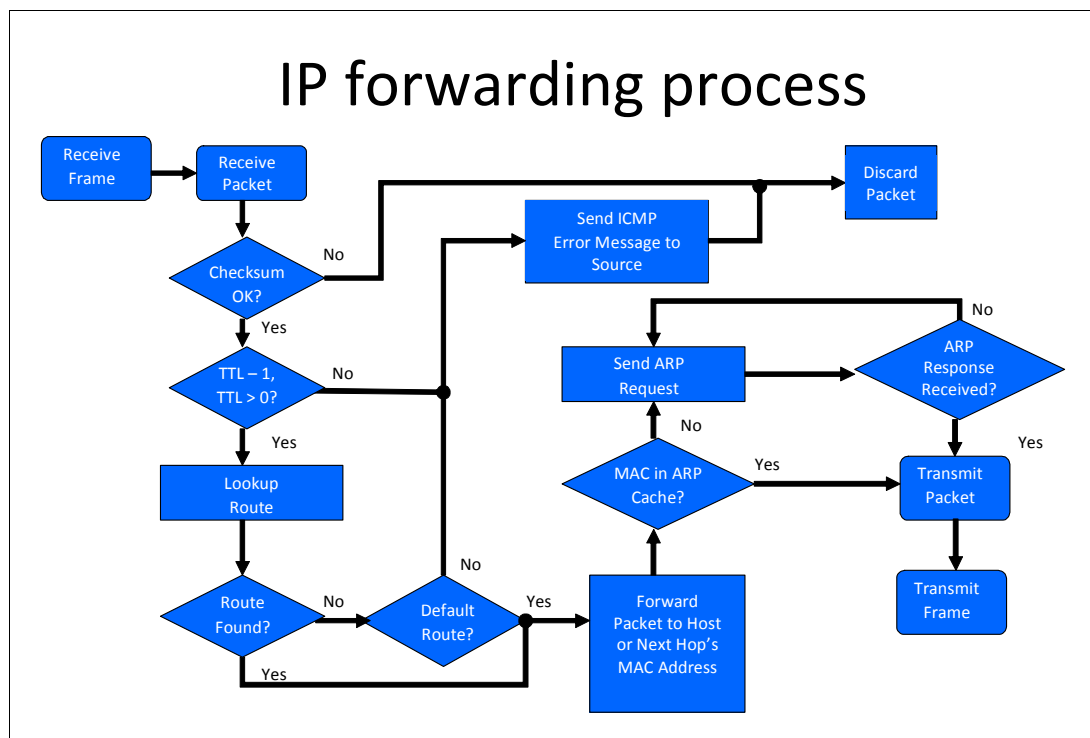


Figure 1-38 IP routing or IP forwarding process

1.5.2 Address Resolution Protocol

Address Resolution Protocol (ARP) gives the MAC addresses of adjacent nodes in the network to the IP, as communications require. IP devices build and maintain a dynamic IP to MAC address mapping in the memory known as an *ARP table* or *ARP cache*.

The MAC address of a target host is learned by the transmitting host sending a request as a Layer 2 broadcast to all devices. This request is known as an *ARP request*. The ARP request contains the IP and MAC address of the requesting device for the response to be sent.

The device with the target IP address responds to the request with the target's MAC address. This response is known as an *ARP reply*. The sending station receives the ARP reply and updates the ARP cache. Now the sending device has the necessary information to send the packet to the target device.

1.5.3 IPv4 addressing

IPv4 addresses are 32 bits in length and are used to logically address devices at the Layer 3 level in a network. The 32 bits are represented in a “dotted quad” notation in which the decimal value of groups of 8 bits of the 32 bits address are separated by decimals, as illustrated in Figure 1-39 on page 35.

IPv4 addresses

- Unique, 32-bit identification for device (NIC, router, etc)
- Commonly written as four decimal numbers
- Dotted quad notation
- Each quad number is the decimal of 8 bits

Binary:

11000000101010000100011001111101

Hex:

C0 A8 46 7D

192 **168** **70** **125**
11000000 · 10101000 · 01000110 · 01111101

Figure 1-39 IPv4 addresses

IP address classes

IPv4 addresses were initially assigned to users on a class basis. Each class provides a different number of usable addresses, as shown in Table 1-3.

Table 1-3 IPv4 Address Classes

IP class	Address range	Subnet mask	Number of host addresses
A	1.x.x.x - 127.x.x.x	255.0.0.0	16,777,214
B	128.0.x.x - 191.255.x.x	255.255.0.0	65534
C	192.0.0.x - 223.255.255.x	255.255.255.0	254
D	224.0.0.0 - 239.255.255.255	Multicast	n/a
E	240.0.0.0 - 255.255.255.255	Reserved	n/a

IPv4 address consumption: In the 1980s, fear of consuming all of the IPv4 addresses available in the Internet fueled several ideas as to how to overcome the limitation of only having approximately 4 billion IPv4 address available for use. Some solutions were implemented to provide methods to more efficiently use the IPv4 address space.

Classless Inter-Domain Routing

Classless Inter-Domain Routing (CIDR) was introduced in 1993 as a replacement to the classfull IP address syntax. CIDR provides greater flexibility in dividing IP address ranges into separate networks and allows for a more efficient use of the IPv4 addresses. IP address blocks are denoted by the IP address and the number of bits in the network mask (10.69.71.91/24).

Subnet Mask

An IP subnet mask is a 32-bit mask used in IPv4 to identify the network portion of the IP address. The remaining bits are the host portion of the address, as shown in Figure 1-40.

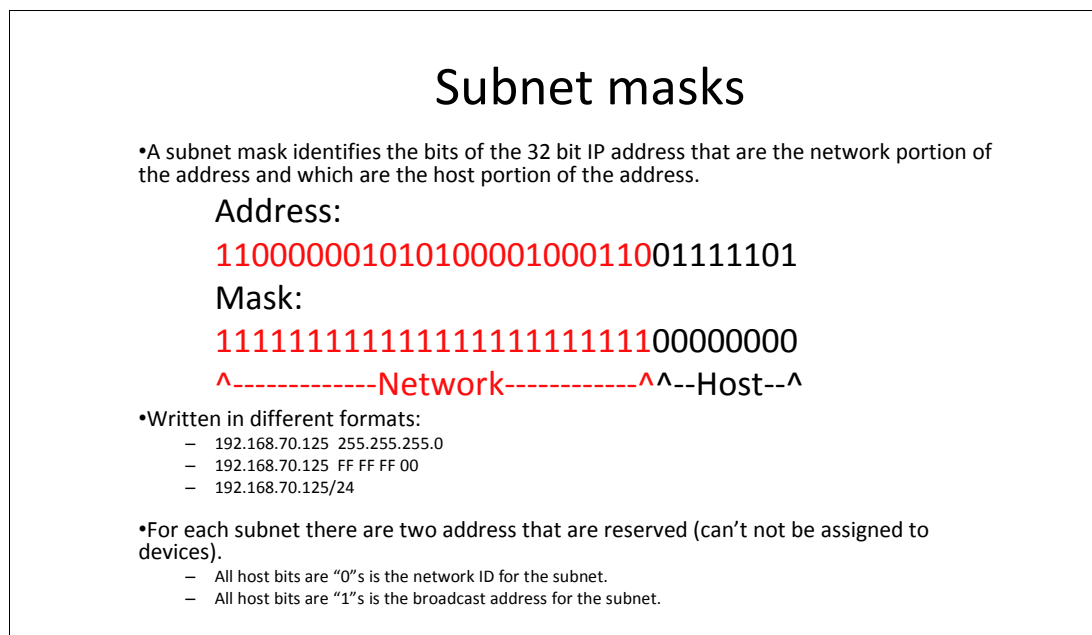


Figure 1-40 Subnet mask

When an IP device communicates with another IP host, the device compares the bits of the local IP address masked by the subnet mask with the bits of the target IP address masked by the same subnet mask. If the masked bits of the two addresses match, the target address is on the same subnet as the sending device; alternately, if they do not match, the target device is on a remote subnet.

In each subnet, two addresses are reserved that cannot be assigned as host addresses.

- **Network ID:** The network ID for a subnet is the address in which hosts bits of the masked address are all *zeros*. The network ID is the first address in the subnet. The network ID is used to reference the subnet in routing tables.
- **Broadcast address:** The broadcast address for a subnet is the address in which hosts bits of the masked address are all *ones*. The broadcast address is the last address in the subnet.

1.5.4 IPv6 addressing

IPv6 was adopted in 1994 by the Internet Engineering Task Force (IETF) as IPng or IP Next Generation and later became known as IPv6. IPv6 has many differences from IPv4. One of the most significant and easily recognizable differences is the IP address, which we illustrate in Figure 1-41 on page 37.

IPv6 addressing

- 128-bit address, written as 8 groups of 4 hexadecimal digits with each group separated by colons.

2001:0db8:0000:0000:0000:0000:c0a8:467d
^----Network Prefix----^ ^-----Host-----^

- 64 bit network prefix portion and 64 bit host portion
- Acceptable notations:
 - 2001:0db8:0000:0000:0000:0000:c0a8:467d
 - 2001:0db8:0:0:0:0:c0a8:467d
 - 2001:0db8::0000:0000:0000:c0a8:467d
 - 2001:db8::c0a8:467d

Figure 1-41 IPv6 addresses

For complete details about IPv6 addressing, refer to IETF RFC 4291 at:

<http://tools.ietf.org/html/rfc4291>

Route table

A route table is a list of the known subnets in a network and the desired path to reach each subnet. The tables contain subnets that are learned from local interfaces, static routes, or learned routes from routing protocols. For each subnet, the route table contains the IP address of the next router in the path, if the subnet is a remote subnet, or the interface, if the subnet is a local subnet to the router.

Routing protocols

Routing protocols provide a dynamic method for updating route tables in the routers of an IP network. If routing protocols are not used, the routes must be manually configured using static routes. There are many routing protocols used in the industry.

Some types of common routing protocols are:

Routing Information Protocol (RIP)

A distance-vector routing protocol. RIP uses hop count for path selection. The path with the fewest number of routers is the preferred path.

Open Shortest Path First (OSPF)

A link-state protocol. OSPF uses path cost for path selections. A designated router runs Dijkstra's algorithm to calculate the shortest path tree for each OSPF Area.

Border Gateway Protocol (BGP)

A path-vector protocol. BGP uses path, rule sets, and network policies for path selection. It supports CIDR and uses route aggregation to decrease the size of routing tables.

Default gateway

A *default gateway* is the device on a subnet that passes traffic from the local subnet to devices on a remote subnet. When a packet's destination IP address is determined to be on a remote subnet, the packet is sent to the default gateway for routing of the packet along the path to the destination device.

Default route

A *default route* is a parameter that is configured in a router that indicates where to forward packets with destination subnets that are not contained in the route table of the router.

1.6 Firewalls

A *firewall* is a barrier between a trusted environment or network and an untrusted environment or network. The firewall blocks malicious or unauthorized activity from the untrusted side of the firewall from reaching devices on the trusted side of the firewall, as illustrated in Figure 1-42.

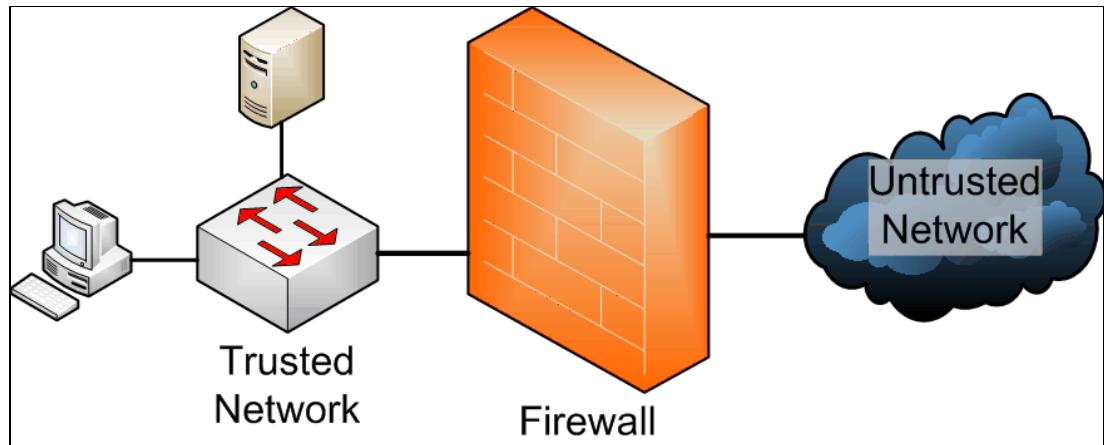


Figure 1-42 Firewalls

Firewalls can be implemented in various forms but are commonly described as *personal firewalls* or *network firewalls*.

1.6.1 Personal firewalls

Personal firewalls are typically implemented as an application that is running on a computer. The application runs in the background, monitors all traffic flow in and out of the computer, and blocks unauthorized traffic.

1.6.2 Network firewalls

Network layer firewalls function in the TCP/IP layer of the network and are categorized as a *statefull firewall* or a *stateless firewall*.

Statefull firewalls

A statefull firewall monitors and maintains the “state” of active, allowed traffic flows through the firewall. If traffic is received that does not match a current session and the state of the session, the firewall takes an appropriate action as defined by the rules in the firewall.

Stateless firewalls

A stateless firewall is not concerned with the state of traffic flows, but it simply filters traffic based on elements of the individual IP packets, such as source or destination IP address, destination port number, or other information that is contained within each packet.

Stateless firewalls versus statefull: Stateless firewalls typically require less resources and make forwarding decisions quicker than statefull firewalls as the state of data flows are not maintained.



Introduction to the IBM j-type Ethernet appliances

In this chapter, we introduce the IBM j-type Ethernet appliance portfolio, which is added to the IBM product line through an original equipment manufacturer (OEM) agreement with Juniper Networks.

This introductory chapter discusses the features of the:

- ▶ IBM Ethernet Appliance J34S and IBM Ethernet Appliance J36S
- ▶ IBM Ethernet Appliance J56S and IBM Ethernet Appliance J58S
- ▶ Junos Operating System

For detailed information about all of the topics discussed in this chapter, refer to the appropriate documentation listed in 2.4, “More information” on page 68.

2.1 IBM Ethernet Appliance J34S and IBM Ethernet Appliance J36S

The IBM Ethernet Appliance J34S and J36S systems are high-performance, scalable security systems for your data center's network infrastructure.

Figure 2-1 shows the IBM Ethernet Appliance J34S.



Figure 2-1 IBM Ethernet Appliance J34S

Figure 2-2 shows the IBM Ethernet Appliance J36S.



Figure 2-2 IBM Ethernet Appliance J36S

If you are familiar with the Juniper Networks products, Table 2-1 on page 43 provides a cross-reference between the IBM j-type s-series Ethernet appliances and the models that Juniper Networks offers.

Table 2-1 IBM j-type s-series to Juniper Networks model cross-reference

IBM description	IBM Machine type and model	Juniper networks model
IBM Ethernet Appliance J34S	4274-S34	SRX3400
IBM Ethernet Appliance J36S	4274-S36	SRX3600

The IBM Ethernet Appliance J34S and J36S are based on an innovative Dynamic Services Architecture that resets the bar in price and performance for enterprise environments. Each multi-services appliance can support near-linear scalability with each additional Services Processing Card (SPC), enabling the J36S to support up to 30 Gbps of firewall throughput, 2.25million concurrent sessions, and 175,000 new VPN connections per second. The appliances offer denial of service (DoS), network address translation (NAT), virtual private network (VPN) support and quality of Service (QoS). The SPCs are designed to support a wide range of services, enabling future support of new capabilities without the need for service-specific hardware. Using SPCs on all services ensures that there are no idle resources based on specific services in operation—maximizing hardware utilization. Market-leading flexibility and price/performance of the IBM Ethernet Appliance J34S and J36S come from the modular architecture. The IBM J34S and IBM J36S are next-generation multi services appliances that deliver leading scalability and service integration in a midsize form factor. These appliances are suited for medium to large enterprise networks including:

- ▶ Campus and Enterprise server farms and data centers
- ▶ Aggregation of departmental or segmented security devices
- ▶ Securing managed services and core networking infrastructure

The IBM J34S and IBM J36S appliances can be equipped with a flexible number of I/O cards (IOCs), network processing cards (NPCs) and service processing cards (SPCs). With these cards, the system can be configured to support the ideal balance of performance and port density so that you can tailor each deployment of the IBM Ethernet Appliance J34S and the IBM Ethernet Appliance J36S to specific network requirements. With this flexibility, the J36S can be configured to support more than 100 Gbps interfaces with choices of Gigabit Ethernet or 10-Gigabit Ethernet ports, firewall performance from 10 to 30 Gbps, and services processing to match specific business needs.

The switch fabric employed in the IBM J34S and the IBM J36S Ethernet Appliances enables the scalability of SPCs, NPCs and IOCs. Supporting up to 320 Gbps of data transfer, the fabric enables maximum processing and I/O capability available in any particular configuration. This level of scalability and flexibility enables uninterrupted expansion and growth of the network infrastructure without the security solution being a barrier. The flexibility of the IBM Ethernet Appliance J34S and IBM Ethernet Appliance J36S extends beyond the innovation and proven benefit of the dynamic services architecture. Enabling the installation of SPCs on both the front and the back of the J34S and J36S, the mid-plane design enables market-leading flexibility and scalability. By doubling the number of SPCs supported in half the rack space needed, the J34S and J36S offer not only the underlying architectural innovation but also innovative physical design.

The feature integration on the IBM Ethernet Appliance J34S and IBM Ethernet Appliance J36S is enabled by the Junos operating system. The J34S and J36S are equipped with a robust list of features that include firewall, Internet Protocol Security (IPsec) VPN, DoS, NAT, and QoS. In addition, incorporating the various features under a single OS greatly optimizes the flow of traffic through the multi services appliance. With the Junos OS, the IBM Ethernet Appliance J34S and IBM Ethernet Appliance J36S enjoy the benefit of a single source OS,

single release train, and one architecture that is also available across IBM j-type e-series switches and IBM j-type m-series routers.

J34S description

The IBM Ethernet Appliance J34S can support up to 2.25 million concurrent sessions, 20 Gbps firewall or 6 Gbps of IPsec VPN, along with up to 175,000 new connections per second. The J34S is ideally suited for securing and segmenting enterprise data center network infrastructures and aggregation of various security solutions. The capability to support unique security policies per zones and its ability to scale with the growth of the network make the J34S an ideal deployment for small-to-mid sized server farms or hosting sites.

J36S description

The IBM Ethernet Appliance J36S is a market leading security solution that supports up to 2.25 million concurrent sessions, 30 Gbps firewall, and a 10 Gbps of IPsec VPN along with up to 175,000 new connections per second. Equipped with the full range of security features, the J36S is ideally suited for securing medium-to-large enterprise data centers, hosted or co-located data centers, or securing next-generation enterprise services and applications. It can also be deployed to secure cloud provider infrastructures where multi tenancy is a requirement. The scalability and flexibility of the J36S multi services appliance make it ideal for consolidating mature security appliances in densely populated data centers, growing data centers, and cloud computing environments.

Line cards

IBM Ethernet Appliance J34S and IBM Ethernet Appliance J36S support a variety of line cards.

Service processing cards

IBM Ethernet Appliance J34S and IBM Ethernet Appliance J36S service processing cards act as the “brains” of the systems and are designed to process all available services on the appliance. By eliminating the need for dedicated hardware for specific services or capabilities, there are no instances in which any piece of hardware is taxed to the limit while other hardware sits idle. All of the processing capabilities of the SPCs support any and all services and capabilities of the appliance. The same SPCs are supported on both the IBM Ethernet J34S and IBM Ethernet Appliance J36S. Figure 2-3 shows the SPC.

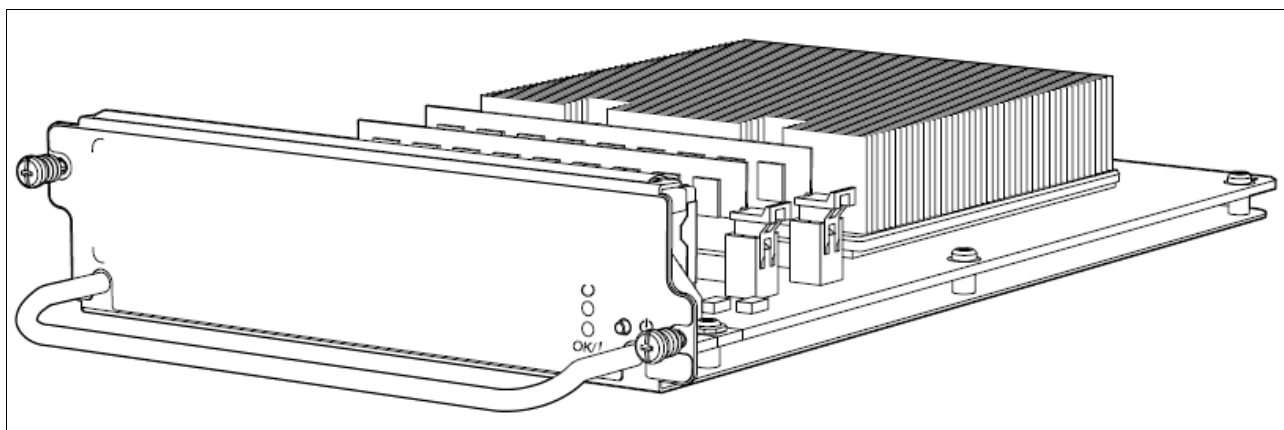


Figure 2-3 Service processing card

Note: A minimum of one NPC and one SPC is required for proper system functionality.

I/O cards

In addition to supporting an ideal mix of built-in copper, small form-factor pluggable transceiver (SFP), and high availability (HA) ports, the IBM Ethernet Appliance J34S and IBM Ethernet Appliance J36S allows the greatest I/O port density of any comparable offering in the same class. The J34S and J36S can be equipped with one or several IOCs, each supporting either 16 x 1 copper or fiber, Gigabit Ethernet, or 2 x 10 Gigabit XFP Ethernet. With the flexibility to provide multiple IOCs, the J34S and J36S can be equipped to support an ideal balance between interfaces and processing capabilities. Figure 2-4 shows the IOC flex modules.

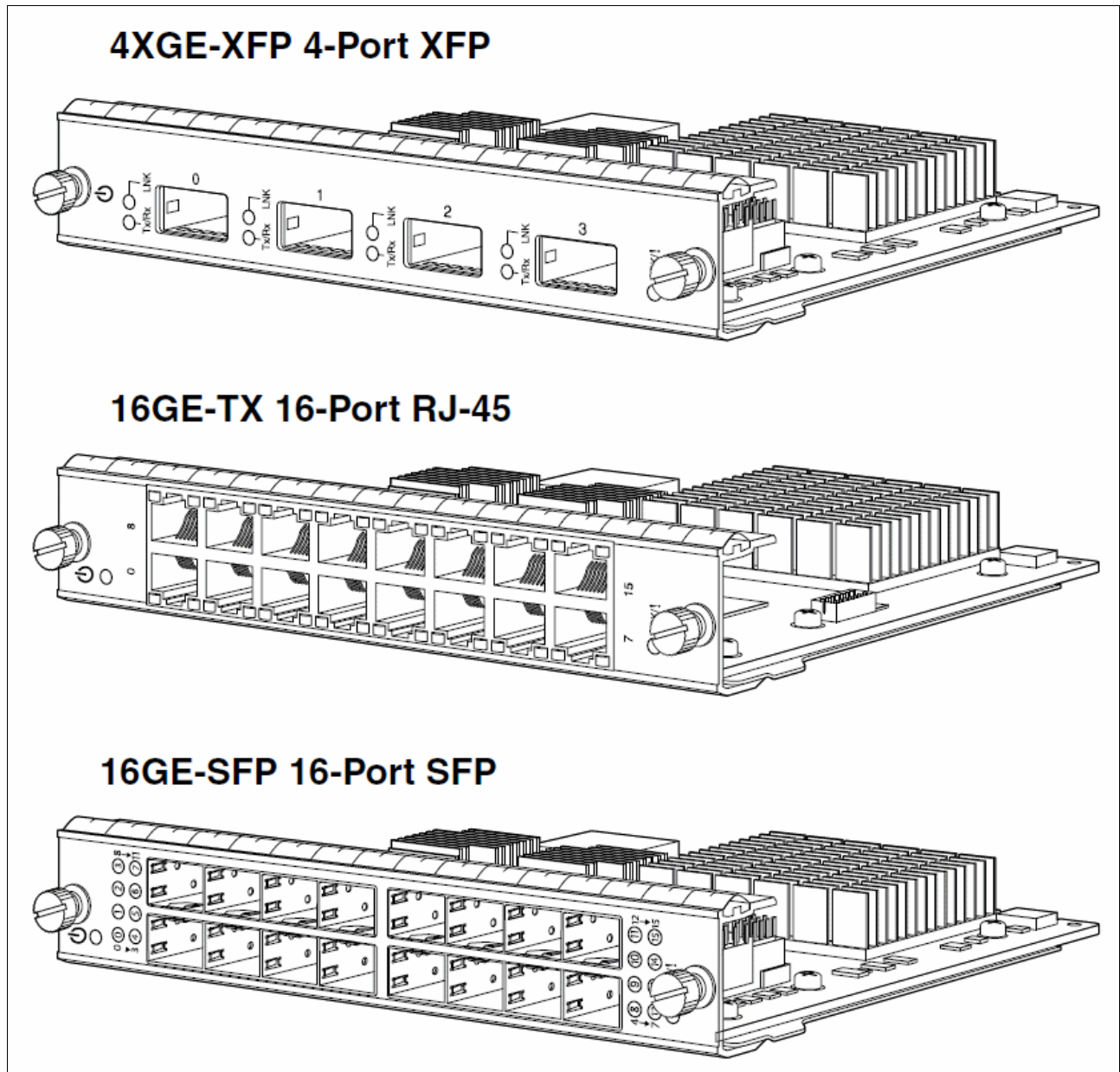


Figure 2-4 IOC flex modules

Network processing cards

To ensure maximum processing performance and flexibility, the IBM Ethernet Appliance J34S and IBM Ethernet Appliance J36S utilize NPCs to distribute in bound and out bound traffic to

the appropriate SPCs and IOCs, apply QoS, and enforce DoS or Distributed Denial of Service (DDoS) protections. The J36S can be configured to support one to three NPCs, while the J34S can be configured to support one or two NPCs. Providing additional NPCs to the appliance allows organizations to tailor the solution to fit their specific performance requirements. Figure 2-5 shows the NPC.

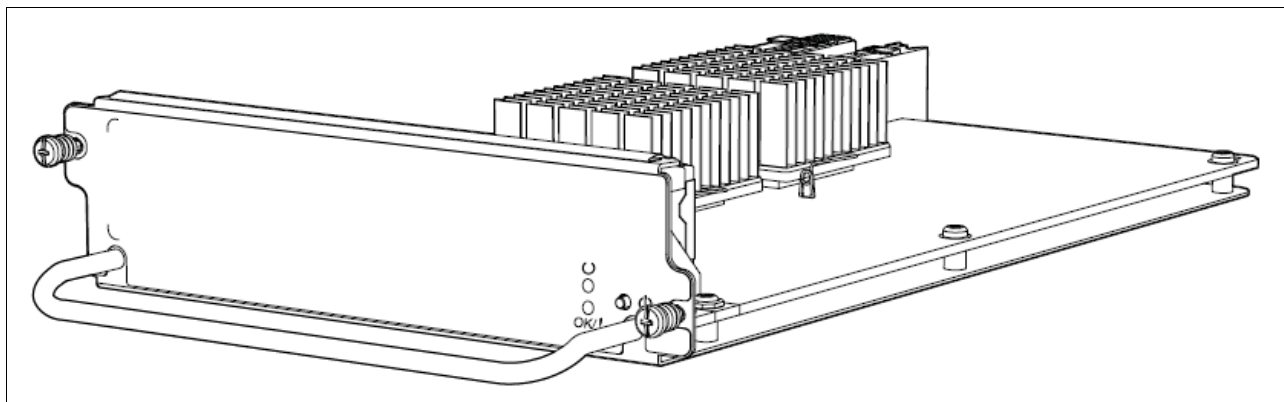


Figure 2-5 Network processing card

Common architectural components

The IBM Ethernet Appliance J34S and IBM Ethernet Appliance J36S share many architectural elements and characteristics with the IBM J5xS series security offerings. The J34S and J36S can accommodate 2.25 million concurrent sessions and 175,000 new session connections per second with a maximum of 40,000 security policies and no restriction of users supported. Both the IBM Ethernet Appliance J34S and IBM Ethernet Appliance J36S run the Junos Software operating system, enabling customers to quickly and cost effectively “turn on” new capabilities as dedicated or multiple services tightly integrate into the operating system. The J34S and J36S also position customers for long-term growth through the Junos Software’s capacity to easily accommodate new capabilities. In addition, the same service processing cards, I/O cards, and network processing cards can be used for the IBM Ethernet Appliance J34S and IBM Ethernet Appliance J36S. The J34S and J36S also come with fixed I/O ports (8 10/100/1000 + 4 SFPs).

Common features

- ▶ Multi services security platform delivers the performance and flexibility to protect high-speed network environments.
- ▶ Scalable performance provides a simple and cost-effective solution to take advantage of new services with appropriate processing.
- ▶ System and network resiliency provides data center-class hardware design and proven OS for reliable network deployments.
- ▶ High availability interfaces to help achieve resiliency necessary to meet the critical demands of enterprise data centers.
- ▶ Interface flexibility offers variable I/O configuration and independent I/O scalability to meet the needs of any particular network requirements.
- ▶ Network segmentation features provide security zones, VLANs, and virtual routers to tailor unique security policies to isolate guests and regional servers or databases.
- ▶ Runs on the modular, fault-tolerant Junos Software operating system.
- ▶ Robust routing engine provides physical and logical separation to data and control planes, enabling the consolidation of routing and security devices.

- Comprehensive threat management features on Junos software—including multi-gigabit firewall, IPsec VPN, DoS, and other services offers integration to protect enterprise networks.

Specifications

The specifications for the IBM Ethernet Appliance J34S and IBM Ethernet Appliance J36S are listed in the following tables. Table 2-2 lists the firewall performance specifications of the IBM Ethernet Appliance J34S and IBM Ethernet Appliance J36S.

Table 2-2 Firewall performance specifications

At a glance	IBM Ethernet Appliance J34S (4274-S34)	IBM Ethernet Appliance J36S (4274-S36)
Firewall performance	20 Gbps (max)	30 Gbps (max)
Firewall performance (IMIX)	8 Gbps	18 Gbps
Firewall packets per second (64 bytes)	3 Mpps	6 Mpps
Maximum AES256+SHA-1 VPN performance	6 Gbps	10 Gbps
Maximum 3DES+SHA-1 VPN performance	6 Gbps	10 Gbps
Maximum IPS performance (NSS 4.2.1)	6 Gbps	10 Gbps
Maximum concurrent sessions	2.25 million	2.25 million
New sessions/second, (sustained, TCP, three-way)	175,000	175,000
Maximum security policies	40,000	40,000
Maximum user supported	Unrestricted	Unrestricted

Table 2-3 list the network connectivity and processing scalability.

Table 2-3 Network connectivity and processing scalability

At a glance	IBM Ethernet Appliance J34S (4274-S34)	IBM Ethernet Appliance J36S (4274-S36)
Network connectivity		
Fixed I/O	8 10/100/1000 + 4 SFP	8 10/100/1000 + 4 SFP
	16 x 1 10/100/1000 copper	16 x 1 Gigabit Ethernet SFP
LAN interface options	16 x 1 Gigabit Ethernet SFP	16 x 1 Gigabit Ethernet SFP
	2 x 10-Gigabit Ethernet XFP	2 x 10-Gigabit Ethernet XFP
Maximum available slots for IOCs	Four (front slots)	Six (front slots)
Processing scalability		
Maximum available slots for SPCs	Up to four SPCs supported per chassis (any slot)	Up to seven SPCs supported per chassis (any slot)

At a glance	IBM Ethernet Appliance J34S (4274-S34)	IBM Ethernet Appliance J36S (4274-S36)
Maximum available slots for NPCs	Up to two NPCs supported per chassis (three rear slots)	Up to three NPCs supported per chassis (three rear-right slots)

Table 2-4 provides firewall capabilities for the IBM J34S and IBM J36S appliances.

Table 2-4 Firewall capabilities

At a glance (Firewall)	IBM Ethernet Appliance J34S (4274-S34)	IBM Ethernet Appliance J36S (4274-S36)
Network attack detection	Yes	Yes
DoS and DDoS protection	Yes	Yes
TCP reassembly for fragmented packet protection	Yes	Yes
Brute-force attack mitigation	Yes	Yes
SYN cookie protection	Yes	Yes
Zone-based IP spoofing	Yes	Yes
Malformed packet protection	Yes	Yes

Table 2-5 lists the IPsec VPN capabilities of the IBM J34S and IBM J36S appliances.

Table 2-5 IPsec VPN

At a glance (IPsec VPN)	IBM Ethernet Appliance J34S (4274-S34)	IBM Ethernet Appliance J36S (4274-S36)
Concurrent VPN tunnels	Up to 10,000	Up to 10,000
DES (56-bit), 3DES (168-bit), and AES encryption	Yes	Yes
MD5 and SHA-1 authentication	Yes	Yes
Manual key, IKE, Public Key Infrastructure (PKI) (X.509)	Yes	Yes
Perfect forward secrecy (DH groups)	1,2,5	1,2,5
Prevent replay attack	Yes	Yes
Remote access VPN	Yes	Yes
Redundant VPN gateways	Yes	Yes

Table 2-6 on page 49 provides the specifications for NAT, both Destination Network Address Translation and Source Network Address Translation, within the IBM J34S and IBM J36S appliances.

Table 2-6 NAT

At a glance	IBM Ethernet Appliance J34S (4274-S34)	IBM Ethernet Appliance J36S (4274-S36)
Destination Network Address Translation		
Destination NAT with PAT	Yes	Yes
Destination NAT within same subnet as ingress interface IP	Yes	Yes
Destination addresses and port numbers to one single address and a specific port number (M:1P)	Yes	Yes
Destination addresses to one single address (M:1)	Yes	Yes
Destination addresses to another range of addresses (M:M)	Yes	Yes
Source Network Address Translation		
Static Source NAT – IP-shifting DIP	Yes	Yes
Source NAT with PAT – port-translated	Yes	Yes
Source NAT without PAT – fix-port	Yes	Yes
Source NAT – IP address persistency	Yes	Yes
Source pool grouping	Yes	Yes
Source pool utilization alarm	Yes	Yes
Source IP outside of the interface subnet	Yes	Yes
Interface source NAT – interface DIP	Yes	Yes
Oversubscribed NAT pool with fallback to	Yes	Yes
PAT when the address pool is exhausted	Yes	Yes
Symmetric NAT	Yes	Yes
Allocate multiple ranges in NAT pool	Yes	Yes
Proxy ARP for physical port	Yes	Yes
Source NAT with loopback grouping – DIP loopback grouping	Yes	Yes

Table 2-7 lists the user authentication and access control capabilities of the IBM J34S and IBM J36S appliances.

Table 2-7 User authentication and access control

At a glance (User authentication and access control)	IBM Ethernet Appliance J34S (4274-S34)	IBM Ethernet Appliance J36S (4274-S36)
Built-in (internal) database	Yes	Yes
RADIUS accounting	Yes	Yes
Web-based authentication	Yes	Yes
UAC enforcement point	Yes	Yes

Table 2-8 contains the specifications for PKI, virtualization, routing, and IP address assignment.

Table 2-8 PKI, virtualization, routing, and IP address assignment

At a glance	IBM Ethernet Appliance J34S (4274-S34)	IBM Ethernet Appliance J36S (4274-S36)
PKI Support and PKI certificate requests (PKCS 7 and PKCS 10)	Yes	Yes
Automated certificate enrollment (SCEP)	Yes	Yes
Certificate authorities supported	Yes	Yes
Self-signed certificates	Yes	Yes
Virtualization		
Maximum number of security zones	256	256
Maximum number of virtual routers	256	256
Maximum number of VLANs per interface	4096	4096
Maximum number of L3 sub interfaces	16,384	16,384
Routing		
BGP instances	128	128
BGP peers	2,000	2,000
BGP routes	1,000,000	1,000,000
OSPF instances	256	256
OSPF routes	1,000,000	1,000,000
RIP v1/v2 instances	50	50
RIP v2 table size	30,000	30,000

At a glance	IBM Ethernet Appliance J34S (4274-S34)	IBM Ethernet Appliance J36S (4274-S36)
Dynamic routing	Yes	Yes
Static routes	Yes	Yes
Filter-based forwarding (FBF)	Yes	Yes
Equal-cost multipath (ECMP)	Yes	Yes
Reverse path forwarding (RPF)	Yes	Yes
IP Address Assignment		
Static	Yes	Yes
Dynamic Host Configuration Protocol (DHCP)	Yes	Yes
Internal DHCP server	Yes	Yes
DHCP relay	Yes	Yes

The traffic management and high availability specifications are listed in Table 2-9.

Table 2-9 Traffic management and high availability

At a glance	IBM Ethernet Appliance J34S (4274-S34)	IBM Ethernet Appliance J36S (4274-S36)
Traffic Management QoS		
Maximum bandwidth	Yes	Yes
RFC2474 IP DiffServ in IPv4	Yes	Yes
Filters for CoS	Yes	Yes
Classification	Yes	Yes
Scheduling	Yes	Yes
Shaping	Yes	Yes
Intelligent Drop Mechanisms (WRED)	Yes	Yes
Three-level scheduling	Yes	Yes
Weighted round-robin for each level of scheduling	Yes	Yes
Priority of routing protocols	Yes	Yes
High Availability		
Active/passive, active/active	Yes	Yes
Configuration synchronization	Yes	Yes
Session synchronization for firewall	Yes	Yes
Session failover for routing change	Yes	Yes

At a glance	IBM Ethernet Appliance J34S (4274-S34)	IBM Ethernet Appliance J36S (4274-S36)
Device failure detection	Yes	Yes
Link failure detection	Yes	Yes

Table 2-9 on page 51 lists the specifications for management, administration, and logging, or monitoring of the IBM J34S and IBM J36S appliances.

Table 2-10 Management, administration, and logging, or monitoring

At a glance	IBM Ethernet Appliance J34S (4274-S34)	IBM Ethernet Appliance J36S (4274-S36)
Management		
Web UI (HTTP and HTTPS)	Yes	Yes
Command-line interface (console)	Yes	Yes
Command-line interface (telnet)	Yes	Yes
Command-line interface (SSH)	Yes	Yes
Network and Security Manager version 2008.2 or later	Yes	Yes
Administration		
Local administrator database support	Yes	Yes
External administrator database support	Yes	Yes
Restricted administrative networks	Yes	Yes
Root, admin, and read-only user levels	Yes	Yes
Software upgrades	Yes	Yes
Configuration rollback	Yes	Yes
Logging, or Monitoring		
Structured System Log	Yes	Yes
SNMP (v2/v3)	Yes	Yes
Traceroute	Yes	Yes

Table 2-11 on page 53 lists the specifications for dimensions and power, certifications, and the operating environment.

Table 2-11 Dimensions and power, certifications, and operating environment

At a glance	IBM Ethernet Appliance J34S (4274-S34)	IBM Ethernet Appliance J36S (4274-S36)
Dimensions (W x H x D)	17.5 x 5.25 x 25.5 in 44.5 x 13.3 x 64.8 cm) 3 RU	17.5 x 8.75 x 25.5 in (44.5 x 22.2 x 64.8 cm) 5 RU
Weight	Chassis 32.3 lb (14.7 kg); Fully configured 75 lb (34.1kg)	Chassis 43.6 lb (19.8 kg); Fully configured 115.7 lb (52.6 kg)
Power		
Power supply (AC)	100 to 240 V ac	100 to 240 V ac
Maximum power draw	1,100 W (ac power)	1,750 W (ac power)
Power supply redundancy	1 + 1	2+1/2+2
Certifications		
Safety certifications	Yes	Yes
Electromagnetic compatibility (EMC) certifications	Yes	Yes
Operating Environment		
Operating temperature	32° to 104° F (0° to 40° C)	32° to 104° F (0° to 40° C)
Humidity	5% to 90% non condensing humidity	5% to 90% non condensing humidity

Hardware summary

Here is a summary of features for the IBM j-type Ethernet appliance models J34S and J36S:

- ▶ Fixed I/O 8 10/100/1000 + 4SFP
- ▶ Four (J34S) and six (J36S) IOC slots (front slots)
- ▶ Up to four (J34S) and up to seven (J36S) SPC slots supported per chassis (any slot)
- ▶ Up to two (J34S) and up to three (J36S) NPC slots supported per chassis
- ▶ 20 Gbps (J34S) and 30 Gbps (J36S) maximum firewall performance
- ▶ 2.25 million maximum concurrent sessions
- ▶ 175,000 new sessions/second (sustained, TCP, three-way)
- ▶ 6 Gbps (J34S) and 10 Gbps (J36S) IPsec VPN
- ▶ Active/Passive, Active/Active high availability support
- ▶ 40,000 maximum security policies
- ▶ Unrestricted maximum users supported
- ▶ 3 Mpps (J34S) and 6 Mpps (Js6S) firewall packets per second (64-byte)
- ▶ 1,000 W ac (J34S) and 1,750 W ac (J36S) maximum power draw
- ▶ Junos operating software

2.2 IBM Ethernet Appliance J56S and IBM Ethernet Appliance J58S

The IBM Ethernet Appliance J56S and J58S systems offer high-performance and scalability, ensuring uninterrupted expansion and growth of the network infrastructure while satisfying data center security requirements.

Figure 2-6 shows the IBM Ethernet Appliance J56S. Figure 2-7 on page 55 shows IBM Ethernet Appliance J58S.



Figure 2-6 IBM Ethernet Appliance J56S



Figure 2-7 IBM Ethernet Appliance J58S

For those who have familiarity with the Juniper Networks products, Table 2-12 provides a cross reference between the IBM j-type s-series Ethernet appliances and the models offered by Juniper Networks.

Table 2-12 IBM Ethernet Appliance J56S and J58S

IBM description	IBM Machine type and model	Juniper Networks model
IBM Ethernet Appliance J56S	4274-S56	SRX5600
IBM Ethernet Appliance J58S	4274-S58	SRX5800

Based on the Dynamic Services Architecture, the IBM Ethernet Appliance J56S and J58S provide market-leading scalability. Each multi services appliance can support almost linear scalability with each additional services processing card (SPC) enabling a fully equipped J58S to support:

- ▶ Up to 120 Gbps firewall throughput
- ▶ Up to 10 million concurrent sessions
- ▶ 350,000 new VPN connections per second
- ▶ A range of features, including DoS protection, NAT, VPN support, and QoS

The SPCs are designed to support a wide range of services that enable future support of new capabilities without the need for service-specific hardware. Using SPCs on all services ensures that there are no idle resources based on specific services being used, which maximizes the utilization of equipped hardware. The IBM J56S and IBM J58S are next

generation multi services appliances that deliver leading scalability and service integration in a mid-sized form factor. These appliances are suited for medium-to-large enterprise networks that includes:

- ▶ Cloud and hosting provider data centers
- ▶ Managed service providers
- ▶ Secure core data center infrastructure
- ▶ Large enterprise data centers
- ▶ Aggregate of departmental or segmented security infrastructure

The scalability and flexibility of the IBM Ethernet Appliance J56S and IBM Ethernet Appliance J58S are supported by equally robust interfaces. The J56S and J58S employ a modular approach to interfaces where the appliance can be equipped with a flexible number of I/O cards. With the IOCs sharing the same interface slot as the SPCs, the appliance can be configured to support the ideal balance of processing and I/O. Hence, each deployment of the J56S and J58S can be tailored to specific network requirements. With this flexibility, the J58S can be configured to support more than 400 gigabit ports with choices of Gigabit Ethernet or 10-Gigabit Ethernet.

The switch fabric employed in the IBM J56S and IBM J58S Ethernet Appliances enables the scalability of SPCs and IOCs. Supporting up to 960 Gbps of data transfer, the fabric enables maximum processing and I/O capability available in any particular configuration. This level of scalability and flexibility enables uninterrupted expansion and growth of the network infrastructure without the security solution being a barrier.

The feature integration on the IBM Ethernet Appliance J56S and IBM Ethernet Appliance J58S is enabled by Junos software. By combining the routing heritage of Junos and the security heritage of Screen OS, the J56S and J58S is equipped with a robust list of features that include firewall, DoS, NAT, and QoS. In addition to the benefit of individual features, incorporating the various features under one OS greatly optimizes the flow of traffic through the services appliance. Network traffic no longer needs to be routed across multiple paths, cards, or even disparate operating systems within a single appliance.

The IBM Ethernet Appliance J56S and IBM Ethernet Appliance J58S running Junos also offers data center reliability and resiliency. The J56S and J58S enjoy the benefit of a single source OS, single release train, and one architecture that is also available across IBM j-type e-series switches and IBM j-type m-series routers.

J58S description

The IBM Ethernet Appliance J58S is the market-leading security solution, supporting up to 120 Gbps firewall, 30Gbps IPsec VPN, 10 million concurrent sessions, and 350,000 new VPN connections per second. Equipped with the full range of security features, the J58S is ideally suited for securing large enterprise data centers, hosted or co-located data center, and cloud provider infrastructures. The massive scalability and flexibility of the services appliance makes it ideal for densely consolidated data centers. The service density makes it ideal for cloud computing environments.

J56S description

The IBM Ethernet Appliance J56S uses the same SPCs and IOCs as the J58S and can support up to 60 Gbps firewall or a15 Gbps IPsec VPN, offering up to 9 million concurrent sessions with 350,000 new sessions per second. The J56S is ideally suited for securing enterprise data centers and aggregation of various security solutions. The capability to support unique security policies per zones and scale with the growth of the network infrastructure makes the IBM Ethernet Appliance J56S an ideal deployment.

Line cards

IBM Ethernet Appliance J56S and IBM Ethernet Appliance J58S support a variety of line cards.

Service processing cards

As the “brain” behind the IBM J5xS security series, the SPCs are designed to process all available services on the appliance. Without the need for dedicated hardware for specific services or capabilities, there are no instances in which a piece of hardware is taxed to the limit while other hardware sits idle. All of the processing capabilities of the SPCs are designed to process all configured services on the appliance. The same SPCs are supported on both IBM Ethernet Appliance J56S and IBM Ethernet Appliance J58S. Figure 2-8 shows the SPC.

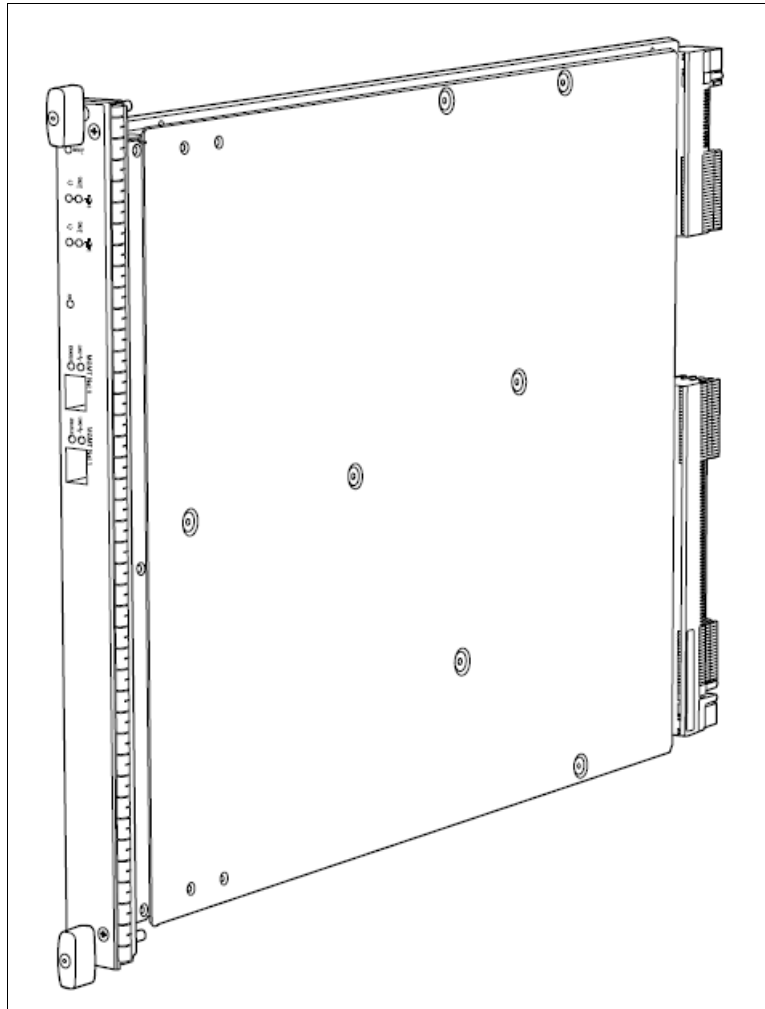


Figure 2-8 J56S and J58S SPC

I/O cards

To provide the most flexible solution, IBM Ethernet Appliance J56S and IBM Ethernet Appliance J58S employ the same modular architecture for SPCs and IOCs. The J56S and J58S can be equipped with one or several IOCs, with each IOC supporting 40 gigabit interfaces (4 x 10-Gigabit Ethernet or 40 x 1Gigabit Ethernet). With the flexibility to install an IOC or an SPC on a given slot, the J56S and J58S can be equipped to support an ideal balance between interfaces and processing capabilities. Figure 2-9 on page 58 shows the Flex IOC with modules installed.

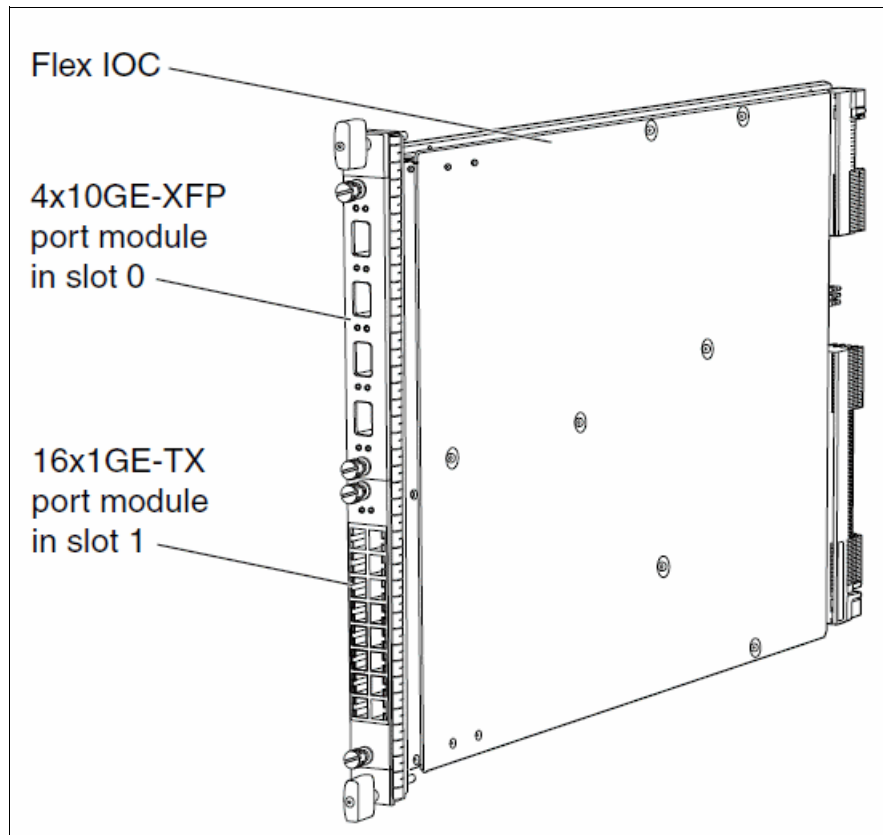


Figure 2-9 Flex IOC with modules installed

Common architectural components

The IBM Ethernet Appliance J56S and IBM Ethernet Appliance J58S share many architectural elements and characteristics of the IBM J3xS series. Based on the Dynamic Services Architecture, the J56S and J58S provide unrivaled scalability. Each appliance can support almost linear scalability, with each additional SPC enabling a fully equipped J58S to support up to 120 Gbps firewall throughput. The SPCs are designed to support a wide range of services enabling future support of new capabilities without the need for service-specific hardware. Using SPCs on all services ensures that there are no idle resources based on specific services being used — maximizing the utilization of equipped hardware. Both the IBM Ethernet Appliance J56S and IBM Ethernet Appliance J58S run the Junos software operating system, enabling you to quickly and cost effectively “turn on” new capabilities because dedicated or multiple services are tightly integrated into the operating system. The J56S and J58S also position customers for long-term growth through the Junos software’s capacity to easily accommodate new capabilities. All major IBM Ethernet Appliance J56S and IBM Ethernet Appliance J58S components are hot-swappable and all central functions are available in redundant configurations designed to provide high operational availability by allowing continuous system operation during maintenance or repairs.

Common features

The common features of the IBM Ethernet Appliance J56S and IBM Ethernet Appliance J58S are:

- ▶ Multi services security platform delivers the performance and flexibility to protect high-speed network environments
- ▶ Scalable performance provides a simple and cost-effective solution to take advantage of new services with appropriate processing

- ▶ System and network resiliency provides data center-class hardware design and proven OS for reliable network deployments
- ▶ High availability interfaces to help achieve resiliency that is necessary to meet the critical demands of enterprise data centers
- ▶ Interface flexibility offers variable I/O configuration and independent I/O scalability to meet the needs of any particular network requirements
- ▶ Network segmentation features provide security zones, VLANs, and virtual routers to tailor unique security policies to isolate guests and regional servers or databases
- ▶ Runs on the modular, fault-tolerant Junos software operating system
- ▶ Robust routing engine provides physical and logical separation to data and control planes, enabling the consolidation of routing and security devices
- ▶ Comprehensive threat management features on Junos software—including multi-gigabit firewall, IPsec VPN, DoS, and other services—offer integration to protect enterprise networks

Specifications

The specifications for the IBM Ethernet Appliance J56S and IBM Ethernet Appliance J58S are listed in the Table 2-13 through Table 2-22 on page 65.

Table 2-13 lists the firewall performance specifications of the IBM Ethernet Appliance J56S and IBM Ethernet Appliance J58S.

Table 2-13 Firewall performance specifications

At a glance (firewall performance)	IBM Ethernet Appliance J56S (4274-S56)	IBM Ethernet Appliance J58S (4274-S58)
Maximum firewall performance	60 Gbps (max)	120 Gbps (max)
Firewall performance (IMIX)	20 Gbps	45 Gbps
Firewall packets per second (64 bytes)	7 Mpps	15 Mpps
Maximum AES256+SHA-1 VPN performance	15 Gbps	30 Gbps
Maximum 3DES+SHA-1 VPN performance	15 Gbps	30 Gbps
Maximum IPS performance (NSS 4.2.1)	15 Gbps	30 Gbps
Maximum concurrent sessions	9 million	10 million
New sessions/second, (sustained, TCP, three-way)	350,000	350,000
Maximum security policies	80,000	80,000
Maximum user supported	Unrestricted	Unrestricted

Table 2-14 on page 60 list the network connectivity and processing scalability.

Table 2-14 Network connectivity and processing scalability

At a glance	IBM Ethernet Appliance J56S (4274-S56)	IBM Ethernet Appliance J58S (4274-S58)
Network Connectivity^a		
Local area network (LAN) interface options	40 x 1 Gigabit Ethernet SFP	40 x 1 Gigabit Ethernet SFP
	4 x 10-Gigabit Ethernet XFP (SR or LR)	4 x 10-Gigabit Ethernet XFP (SR or LR)
	16 x 1 Gigabit Ethernet Flex IOC	16 x 1 Gigabit Ethernet Flex IOC
	4 x 10-Gigabit Ethernet XFP Flex IOC	4 x 10-Gigabit Ethernet XFP Flex IOC
Maximum available slots for IOCs	Five	Eleven
Processing scalability^a		
Maximum available slots for SPCs	Five	Eleven
SPC Options	Dual CPU with 8 GB Total Memory	Dual CPU with 8 GB Total Memory

a. IBM Ethernet Appliance J56S and J58S use the same SPCs and IOCs.

Table 2-15 provides firewall capabilities for the IBM J56S and IBM J58S appliances.

Table 2-15 Firewall capabilities

At a glance	IBM Ethernet Appliance J56S (4274-S56)	IBM Ethernet Appliance J58S (4274-S58)
Firewall		
Network attack detection	Yes	Yes
DoS and DDoS protection	Yes	Yes
TCP reassembly for fragmented packet protection	Yes	Yes
Brute-force attack mitigation	Yes	Yes
SYN cookie protection	Yes	Yes
Zone-based IP spoofing	Yes	Yes
Malformed packet protection	Yes	Yes

Table 2-16 lists the IPsec VPN capabilities of the IBM J56S and IBM J58S appliances.

Table 2-16 IPsec VPN

At a glance	IBM Ethernet Appliance J56S (4274-S56)	IBM Ethernet Appliance J58S (4274-S58)
IPsec VPN		
Tunnel interfaces	10,000	10,000

At a glance	IBM Ethernet Appliance J56S (4274-S56)	IBM Ethernet Appliance J58S (4274-S58)
DES (56-bit), 3DES (168-bit), and AES encryption	Yes	Yes
MD5 and SHA-1 authentication	Yes	Yes
Manual key, IKE, Public Key Infrastructure (PKI) (X.509)	Yes	Yes
Perfect forward secrecy (DH groups)	1,2,5	1,2,5
Prevent replay attack	Yes	Yes
Remote access VPN	Yes	Yes
Redundant VPN gateways	Yes	Yes

Table 2-17 provides the specifications for NAT, both Destination Network Address Translation and Source Network Address Translation, within the IBM J56S and IBM J58S appliances.

Table 2-17 NAT

At a glance	IBM Ethernet Appliance J56S (4274-S56)	IBM Ethernet Appliance J58S (4274-S58)
Destination Network Address Translation		
Destination NAT with PAT	Yes	Yes
Destination NAT within same subnet as ingress interface IP	Yes	Yes
Destination addresses and port numbers to one single address and a specific port number (M:1P)	Yes	Yes
Destination addresses to one single address (M:1)	Yes	Yes
Destination addresses to another range of addresses (M:M)	Yes	Yes
Source Network Address Translation		
Static Source NAT – IP-shifting DIP	Yes	Yes
Source NAT with PAT – port-translated	Yes	Yes
Source NAT without PAT – fix-port	Yes	Yes
Source NAT – IP address persistence	Yes	Yes
Source pool grouping	Yes	Yes
Source pool utilization alarm	Yes	Yes

At a glance	IBM Ethernet Appliance J56S (4274-S56)	IBM Ethernet Appliance J58S (4274-S58)
Source IP outside of the interface subnet	Yes	Yes
Interface source NAT – interface DIP	Yes	Yes
Oversubscribed NAT pool with fallback to	Yes	Yes
PAT when the address pool is exhausted	Yes	Yes
Symmetric NAT	Yes	Yes
Allocate multiple ranges in NAT pool	Yes	Yes
Proxy ARP for physical port	Yes	Yes
Source NAT with loopback grouping – DIP loopback grouping	Yes	Yes

Table 2-18 lists the user authentication and access control capabilities of the IBM J34S and IBM J36S appliances.

Table 2-18 User authentication and access control

At a glance	IBM Ethernet Appliance J56S (4274-S56)	IBM Ethernet Appliance J58S (4274-S58)
User authentication and access control		
Built-in (internal) database	Yes	Yes
RADIUS accounting	Yes	Yes
Web-based authentication	Yes	Yes
UAC enforcement point	Yes	Yes

Table 2-19 contains the specifications for PKI, virtualization, routing, and IP address assignment.

Table 2-19 PKI, virtualization, routing, and IP address assignment

At a glance	IBM Ethernet Appliance J56S (4274-S56)	IBM Ethernet Appliance J58S (4274-S58)
PKI		
PKI support and PKI certificate requests (PKCS 7 and PKCS 10)	Yes	Yes
Automated certificate enrollment (SCEP)	Yes	Yes
Certificate authorities supported	Yes	Yes

At a glance	IBM Ethernet Appliance J56S (4274-S56)	IBM Ethernet Appliance J58S (4274-S58)
Self-signed certificates	Yes	Yes
Virtualization		
Maximum number of security zones	256	512
Maximum number of virtual routers	500	500
Maximum number of VLANs per interface	4096	4096
Routing		
BGP instances	128	128
BGP peers	2,000	2,000
BGP routes	1,000,000	1,000,000
OSPF instances	400	400
OSPF routes	1,000,000	1,000,000
RIP v1/v2 instances	50	50
RIP v2 table size	30,000	30,000
Dynamic routing	Yes	Yes
Static routes	Yes	Yes
Policy-based routing	Yes	Yes
Equal-cost multipath (ECMP)	Yes	Yes
Reverse path forwarding (RPF)	Yes	Yes
IP address assignment		
Static	Yes	Yes
Dynamic Host Configuration Protocol (DHCP)	Yes	Yes
Internal DHCP server	Yes	Yes
DHCP relay	Yes	Yes

The traffic management and high availability specifications are listed in Table 2-20.

Table 2-20 Traffic management and high availability

At a glance	IBM Ethernet Appliance J56S (4274-S56)	IBM Ethernet Appliance J58S (4274-S58)
Traffic management QoS		
Maximum bandwidth	Yes	Yes
RFC2474 IP DiffServ in IPv4	Yes	Yes
Filters for CoS	Yes	Yes

At a glance	IBM Ethernet Appliance J56S (4274-S56)	IBM Ethernet Appliance J58S (4274-S58)
Classification	Yes	Yes
Scheduling	Yes	Yes
Shaping	Yes	Yes
Intelligent Drop Mechanisms (WRED)	Yes	Yes
Three-level scheduling	Yes	Yes
Weighted round-robin for each level of scheduling	Yes	Yes
Priority of routing protocols	Yes	Yes
High availability		
Active/passive, active/active	Yes	Yes
Configuration synchronization	Yes	Yes
Session synchronization for firewall and Ipsec VPN	Yes	Yes
Session fail over for routing change	Yes	Yes
Device failure detection	Yes	Yes
Link failure detection	Yes	Yes

Contained in Table 2-21 are all of the specifications for management, administration, and logging, or monitoring of the IBM J56S and IBM J58S appliances.

Table 2-21 Management, administration, and logging, or monitoring

At a glance	IBM Ethernet Appliance J56S (4274-S56)	IBM Ethernet Appliance J58S (4274-S58)
Management		
Web UI (HTTP and HTTPS)	Yes	Yes
Command-line interface (console)	Yes	Yes
Command-line interface (telnet)	Yes	Yes
Command-line interface (SSH)	Yes	Yes
Network and Security Manager version 2008.2 or later	Yes	Yes
Administration		
Local administrator database support	Yes	Yes
External administrator database support	Yes	Yes

At a glance	IBM Ethernet Appliance J56S (4274-S56)	IBM Ethernet Appliance J58S (4274-S58)
Restricted administrative networks	Yes	Yes
Root, admin, and read-only user levels	Yes	Yes
Software upgrades	Yes	Yes
Configuration rollback	Yes	Yes
Logging or Monitoring		
Structured System Log	Yes	Yes
SNMP (v2)	Yes	Yes
Traceroute	Yes	Yes

In Table 2-22, is a listing of the specifications for dimensions and power, certifications, and the operating environment.

Table 2-22 Dimensions and power, certifications, and operating environment

At a glance	IBM Ethernet Appliance J56S (4274-S56)	IBM Ethernet Appliance J58S (4274-S58)
Dimensions (W x H x D)	17.5 x 14 x 23.8 in	17.5 x 27.8 x 23.5 in
	(44.5 x 35.6 x 60.5 cm) 8 RU	(44.5 x 70.5 x 59.7 cm) 16 RU
Weight	Fully configured 180 lb (81.7 kg)	Fully configured 334 lb (151.6 kg)
Power supply (AC)	100 to 240 V ac	200 to 240 V ac
Maximum power draw	2,800 W (ac power)	5,100 W (ac power)
Certifications		
Safety certifications	Yes	Yes
Electromagnetic compatibility (EMC) certifications	Yes	Yes
Operating temperature	32° to 104° F (0° to 40° C)	32° to 104° F (0° to 40° C)
Humidity	5% to 90% non condensing humidity	5% to 90% non condensing humidity

Hardware summary

Here is a summary of features for the IBM j-type Ethernet appliance models J56S and J58S:

- ▶ Five (J56S) and 11 (J58S) slots
- ▶ 60 Gbps (J56S) and 120 Gbps (J58S) maximum firewall performance
- ▶ 9 million (J56S) and 10 million (J58S) maximum concurrent sessions
- ▶ 350,000 new sessions/second (sustained, TCP, three-way)
- ▶ 15 Gbps (J56S) and 30 Gbps (J58S) IPsec VPN
- ▶ Active/Passive, Active/Active high availability support
- ▶ 80,000 maximum security policies
- ▶ Unrestricted maximum users supported

- ▶ 7 Mpps (J56S) and 15 Mpps (J58S) firewall packets per second (64-byte)
- ▶ 2,800 W ac (J56S) and 5,100 W ac (J58S) maximum power draw
- ▶ Junos operating software

2.3 Junos operating system

IBM Ethernet Appliances use Junos software, an advanced network operating system that powers some of the world's largest and most complex networks.

2.3.1 A common network operating system

Junos software is the field-proven operating system powering IBM j-type products within the data center. It enables the consolidation of switching and routing onto a common operating system with feature consistency and interoperability across the entire data center network. Manageability and flexibility of the data center are enhanced to address business needs as they arise and improve data center operations. A common set of tools allows administrators to monitor, administer, and troubleshoot the network, allowing data center operations teams to function more efficiently with less training and providing higher availability for users. Unlike any other networking infrastructure OS on the market, Junos provides one operating system that is enhanced through a single release train and developed upon a common modular architecture—giving enterprises a “1-1-1” advantage.

Figure 2-10 shows that Junos Software runs on the entire IBM j-type infrastructure.

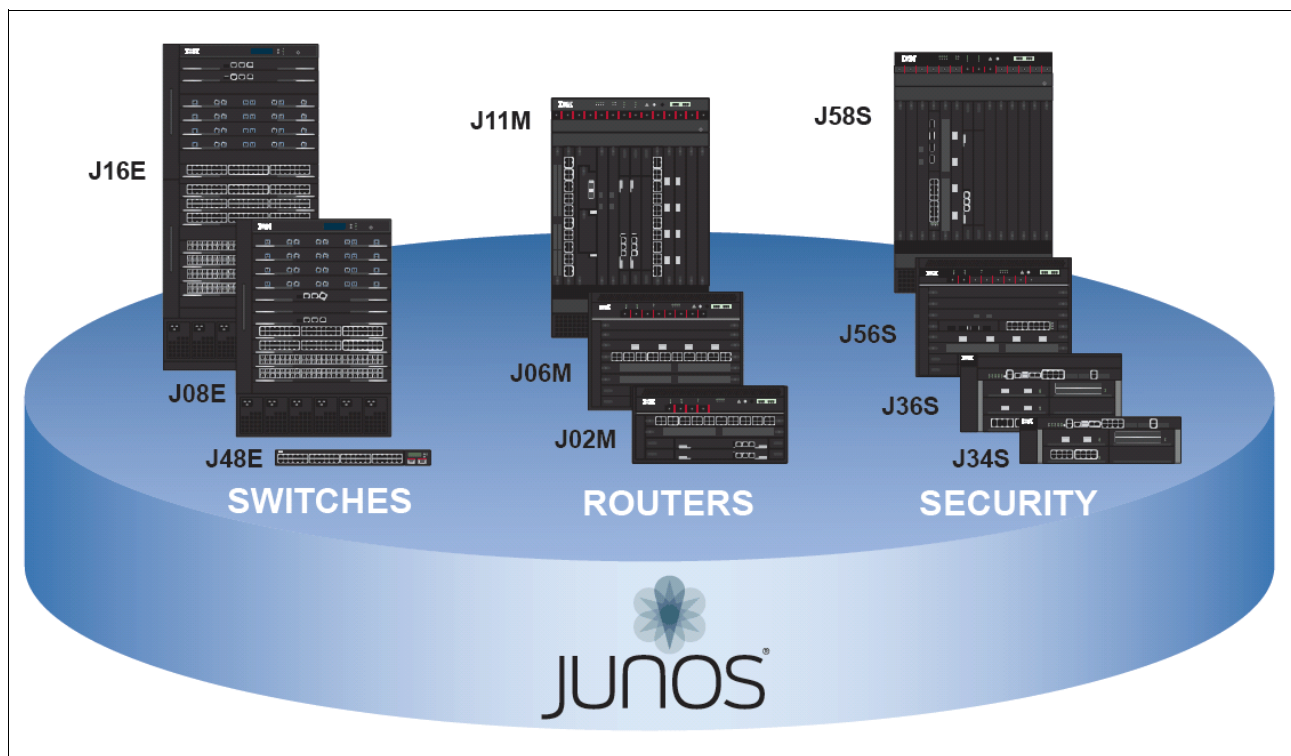


Figure 2-10 Junos software runs on all IBM j-type systems

IBM provides high-performance network devices that create a responsive and trusted environment for accelerating the deployment of services and applications over a single network. Junos software is the foundation of these high-performance networks.

Refer to *IBM j-type Ethernet Switches and Routers Implementation*, SG24-7882 for information about the IBM j-type Ethernet switches and routers that are represented in Figure 2-10 on page 66.

Unlike other complex, monolithic software architectures Junos software incorporates key design and developmental differences to deliver increased network availability, operational efficiency, and flexibility. The key advantages to this approach are:

- ▶ One operating system
- ▶ One software release
- ▶ One modular software architecture

Now we look closer at each of these key advantages.

One operating system

Unlike other network operating systems that share a common name but splinter into many different programs, Junos software is a single, cohesive operating system that is shared across all IBM j-type products. This sharing allows IBM engineers to develop software features one time and share these features across all IBM j-type products simultaneously. Because features are common to a single source, they generally are implemented the same way for all IBM j-type products, thus reducing the training required to learn multiple tools and methods for each product. Because all IBM j-type products use the same code base, interoperability between products is not an issue.

One software release

Each new version of Junos software is released concurrently for all IBM j-type products following a preset quarterly schedule. Furthermore, each new version of software must include all working features released in previous releases of the software and must have no critical regression errors. This discipline ensures reliable operations for the entire release.

One modular software architecture

Although individual modules of the Junos software communicate through well-defined interfaces, each module runs in its own protected memory space, preventing one module from disrupting another. This separation enables the independent restart of each module as necessary. This is in contrast to monolithic operating systems where a malfunction in one module can ripple to other modules and cause a full system crash or restart. This modular architecture then provides for high performance, high availability, device scalability, and security not found in other operating systems.

The Junos software is pre installed on your IBM j-type equipment when you receive it from the factory; therefore, when you first power on the system, all software starts automatically. You simply configure the software so that the system can participate in the network. You can upgrade the Junos software as new features are added or software problems are fixed. You normally obtain new software by downloading the software installation packages from the IBM Support Web page onto your system or onto another system on your local network. You then install the software upgrade onto the IBM j-type system. IBM j-type systems run only binaries supplied by IBM. Each Junos software image includes a digitally signed manifest of executables that are registered with the system only if the signature can be validated. Junos software will not execute any binary without a registered signature. This feature protects the system against unauthorized software and activity that might compromise the integrity of your IBM j-type system.

For additional information about the IBM Junos operating system, refer to Chapter 4, “Junos fundamentals” on page 99.

2.4 More information

For a complete introduction to the IBM j-type portfolio of products and technologies, refer to

<http://www-03.ibm.com/systems/networking/hardware/j-type/index.html>,

You can also refer to the *IBM j-type Data Center Networking Introduction*, SG24-7820.

For more information about the IBM j-type Ethernet appliances and the related hardware, refer to:

- ▶ *IBM Ethernet Appliance J34S Hardware Guide*, GA32-0748
- ▶ *IBM Ethernet Appliance J34S Getting Started Guide*, GA32-0749
- ▶ *IBM Ethernet Appliance J36S Hardware Guide*, GA32-0750
- ▶ *IBM Ethernet Appliance J36S Getting Started Guide*, GA32-0751
- ▶ *IBM Ethernet Appliance J56S Hardware Guide*, GA32-0752
- ▶ *IBM Ethernet Appliance J56S Getting Started Guide*, GA32-0753
- ▶ *IBM Ethernet Appliance J58S Hardware Guide*, GA32-0754
- ▶ *IBM Ethernet Appliance J58S Getting Started Guide*, GA32-0755.

For more information about the Junos software, refer to:

- ▶ *Juniper Web Device Manager for IBM j-type Ethernet Switches and Routers Interface User Guide*, GA32-0688
- ▶ *JUNOS Software Access Privilege Configuration Guide*, GA32-0696
- ▶ *JUNOS Software Broadband Subscriber Management Solutions Guide*, GA32-0709
- ▶ *JUNOS Software Class of Service Configuration Guide*, GA32-0738
- ▶ *JUNOS Software CLI User Guide*, GA32-0697
- ▶ *JUNOS Software Configuration and Diagnostic Automation Guide*, GA32-0679
- ▶ *JUNOS Software Ethernet Routing Engine Media Upgrade Kit*, GA32-0681
- ▶ *JUNOS Software Feature Guide*, GA32-0739
- ▶ *JUNOS Software Hierarchy and RFC Reference*, GA32-0712
- ▶ *JUNOS Software High Availability Configuration Guide*, GA32-0670
- ▶ *JUNOS Software IBM j-type m-series Ethernet Routers Layer 2 Configuration Guide*, GA32-0708
- ▶ *JUNOS Software Installation and Upgrade Guide*, GA32-0695
- ▶ *JUNOS Software Interfaces Command Reference*, GA32-0672
- ▶ *JUNOS Software JUNOScript API Guide*, GA32-0674
- ▶ *JUNOS Software MPLS Applications Configuration Guide*, GA32-0702
- ▶ *JUNOS Software Multicast Protocols Configuration Guide*, GA32-0703
- ▶ *JUNOS Software NETCONF API Guide*, GA32-0678
- ▶ *JUNOS Software Network Interfaces Configuration Guide*, GA32-0706
- ▶ *JUNOS Software Network Management Configuration Guide*, GA32-0698
- ▶ *JUNOS Software Policy Framework Configuration Guide*, GA32-0704
- ▶ *JUNOS Software Routing Protocols and Policies Command Reference*, GA32-0673
- ▶ *JUNOS Software Services Interfaces Configuration Guide*, GA32-0707
- ▶ *JUNOS Software Subscriber Access Configuration Guide*, GA32-0711

- ▶ *JUNOS Software System Basics and Services Command Reference*, GA32-0671
- ▶ *JUNOS Software System Log Messages Reference*, GA32-0675
- ▶ *JUNOS Software VPNs Configuration Guide*, GA32-0705
- ▶ *JUNOScope Software User Guide*, GA32-0670



Initial hardware planning of IBM j-type appliances

In this chapter, we expand on a few of the items that are associated with installing the IBM j-type appliances. We discuss items, such as power requirements, air-flow, racks, and cabling as it pertains to successfully installing an IBM j-type appliance.

This chapter includes the following topics:

- ▶ Power considerations
- ▶ Cabling
- ▶ Cooling system
- ▶ Racks

3.1 Installation overview

In this section, we provide an overview of the installation topics for the IBM j-type s-series systems. For complete information about a particular hardware model and step-by-step procedure of the system installation, refer to the appropriate documents listed in 3.6, “More information” on page 96.

However, there are areas of the installation that we want to discuss in detail here because these areas have a long-term affect on the reliability, resilience, and overall success of the installation. In the following sections of this chapter, we focus on:

- ▶ Power
- ▶ Cooling system
- ▶ Racks
- ▶ Cabling

3.2 Power considerations

The IBM j-type appliances offer a variety of options. Therefore, the number of power supplies required varies by the number of options included with the system and the level of redundancy required for the solution.

In the following sections, we discuss the details of power for the IBM j-type appliances.

3.2.1 AC power circuit breaker requirements for the IBM j-type appliances

Each AC power supply has a single AC appliance inlet located in the chassis directly above the power supply that requires a dedicated AC power feed. We recommend that you use a dedicated customer site circuit breaker rated for 15 A (250 VAC) minimum for each AC power supply, or as required by local code.

3.2.2 Power specifications and requirements for the IBM Ethernet Appliance J34S and J36S

The IBM Ethernet Appliance J34S uses one AC power supply. A second AC power supply can be used to offer redundancy.

The IBM Ethernet Appliance J36S uses two AC power supplies. A third AC power supply, or a third and fourth AC power supply, can be used to offer redundancy.

The power supplies connect to the mid plane, which distributes the various output voltages produced by the power supplies to the Ethernet appliance components, depending on their voltage requirements.

All power supplies are hot-removable and hot-insertable, assuming there is another functioning power supply offering resiliency. Each power supply is cooled by its own internal cooling system.

In the following sections, we discuss the details of power for the IBM J34S and J36S appliances.

IBM Ethernet Appliance J34S and J36S AC power supply overview

The IBM Ethernet Appliance J34S and J36S contains one to four AC power supplies, located at the rear of the chassis in slots PEM0 through PEM3.

Each power supply provides power to all components in the Ethernet appliances. When two, three, or four power supplies are present they share power almost equally within a fully populated system. The two, three, or four AC power supplies provide full power redundancy.

Note: The J36S does not offer full redundancy with only two power supplies.

If one power supply fails or is removed, the remaining power supplies redistribute the electrical load without interruption. The device reassesses the power that is required to support its configuration and issue errors, if the available power is insufficient to support all options that are installed within the system. Figure 3-1 shows the power supply used in both the J34S and the J36S.

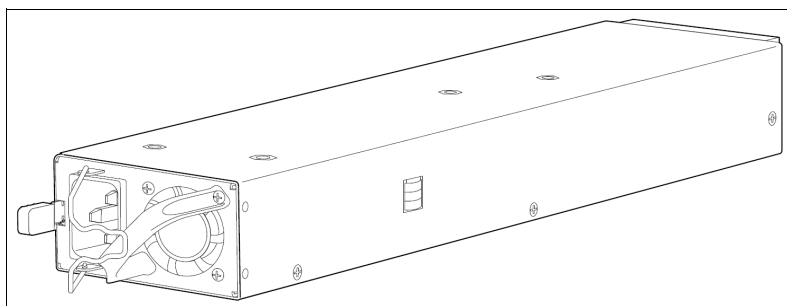


Figure 3-1 J34S and J36S power supply

IBM Ethernet Appliance J34S and J36S AC power supply electrical specifications

Table 3-1 lists the AC power supply electrical specifications.

Table 3-1 J34S and J36S power supply electrical specifications

Parameter	Specification
Maximum output power	1000 W
AC input voltage	100-127 V / 200-240 V
AC input line frequency	50 / 60 Hz
AC input current rating	12.0 A @ 100-127 V / 7 A @ 200-240 V

AC power supply LEDs in the IBM J34S and J36S

Each AC power supply faceplate displays a single LED to indicate the status of the power supply, as shown in Table 3-2 on page 74.

Table 3-2 J34S and J36S Power Supply LED

Color	Status	Condition indicated
Green	On steadily	AC input voltage is present, and both main 12V output and standby 3.3V output are enabled and healthy.
	Blinking	AC input voltage is present, standby 3.3V output is on, but the main 12V output is disabled. This condition usually indicates that the device was powered off at the power button on the SFB front panel. It might also indicate that either the SFB or RE was removed from the services gateway.
Yellow	On steadily	The power supply detected one or more of the following faults: <ul style="list-style-type: none"> ▶ Power supply fan failure ▶ Power supply over-temperature condition ▶ Over-current or under-voltage condition on the standby 3.3V output
	Blinking	The power supply detected one or more of the following faults: <ul style="list-style-type: none"> ▶ Under-voltage condition on the 12V output ▶ Over-voltage condition on the 12V output ▶ Over-current condition on the 12V output

Grounding the IBM Ethernet Appliance J34S and J36S

You ground the devices by connecting a grounding cable to earth ground and then attaching it to the chassis grounding point using two M5 screws. You must provide the grounding cables (the cable lugs are supplied with the device).

To verify that a licensed electrician attached the cable lug provided with the device to the grounding cable:

1. Attach an electrostatic discharge (ESD) grounding strap to your bare wrist and connect the strap to an approved site ESD grounding point. Reference the instructions for your site.
2. Ensure that all grounding surfaces are clean and brought to a bright finish before grounding connections are made.
3. Connect the grounding cable to a proper earth ground.
4. Place the grounding cable lug over the grounding point—a pair of M5 holes to the left of the power supply slots. Figure 3-2 on page 75 shows the grounding points.
5. Detach the ESD grounding strap from the site ESD grounding lug.

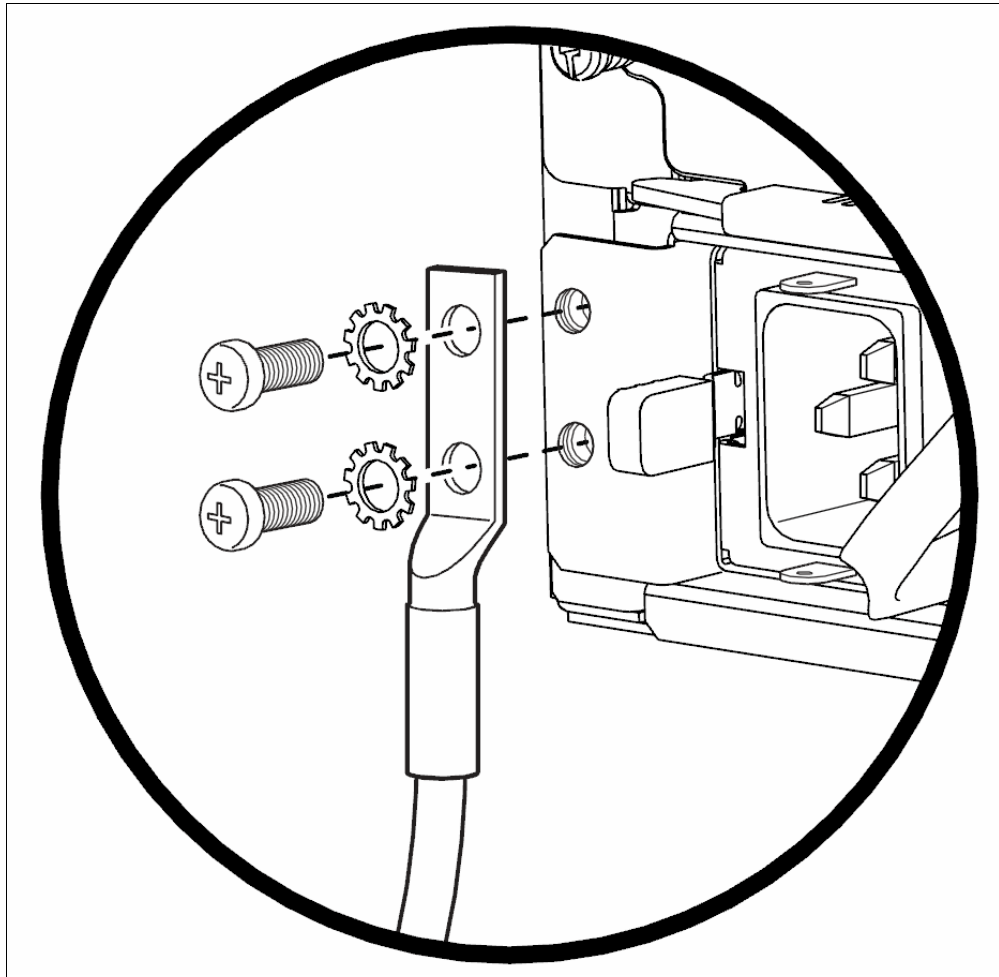


Figure 3-2 j-type grounding lug

AC power, connection, and power cord specifications for the IBM J34S and J36S Appliances

Each AC power supply has a single AC appliance inlet located on the power supply that requires a dedicated AC power feed. Most sites distribute power through a main conduit that leads to frame-mounted power distribution panels, one of which can be located at the top of the rack that houses the device. An AC power cord connects each power supply to the power distribution panel.

The device is not shipped with AC power cords. You must order power cords separately using the model number shown in Table 3-3.

The Ethernet appliance uses detachable AC power cords with C19 appliance couplers at the female end, as described by the international Electrotechnical Commission (IEC) standard 60320.

Table 3-3 AC Power Cord specifications for IBM Ethernet Appliance J34S and J36S

Country	Model number	Plug type
Australia/New Zealand	CBL-PWR-C19S-152-AU	SAA/3/15
China	CBL-PWR-C19S-162-CH	CH2-16P

Country	Model number	Plug type
Continental Europe (except Denmark, Italy, Switzerland, and United Kingdom)	CBL-PWR-C19S-162-EU	CEE 7/7
Italy	CBL-PWR-C19S-162-IT	CEI 23-16/VII
Japan	CBL-PWR-C19S-162-US	NEMA 5-15P
	CBL-PWR-C19S-162-JPL	NEMA L5-15P
North America	CBL-PWR-C19S-151-US15	NEMA 5-15P
	CBL-PWR-C19S-162-US	NEMA L5-15P
	CBL-PWR-C19S-162-JPL	NEMA L5-15P
United Kingdom	CBL-PWR-C19S-132-UK	BS89/13

The plug at the male end of the power cord fits into the power source receptacle that is standard for your geographical location. Table 3-3 on page 75 provides specifications and Figure 3-3 depicts the plug on the AC power cord that is available for each country or region.

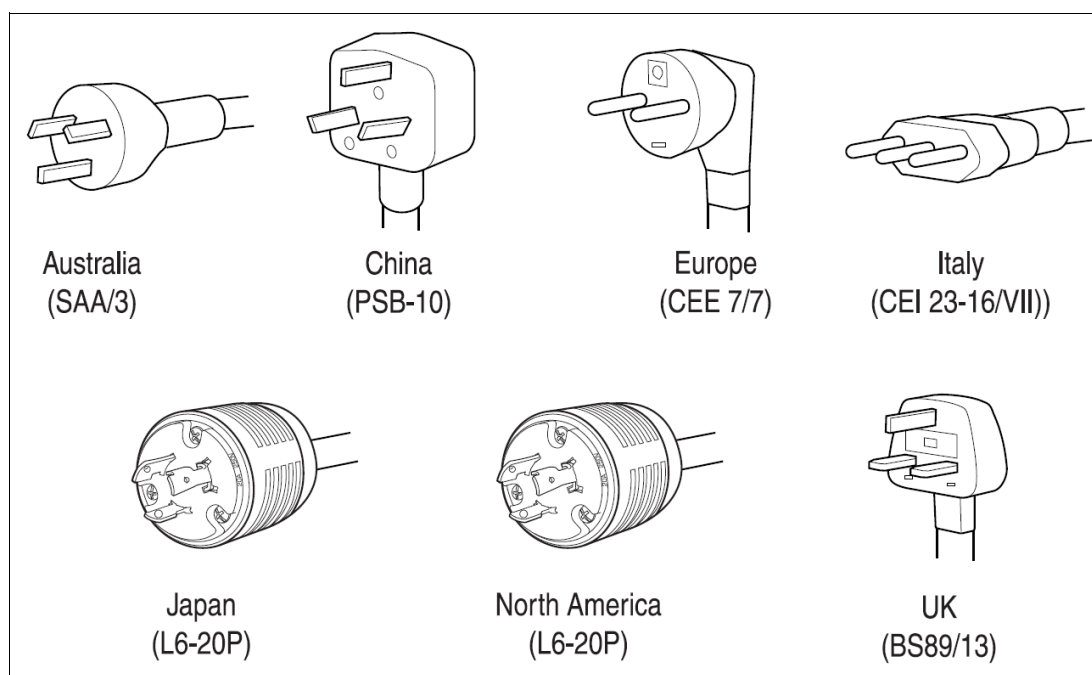


Figure 3-3 J34S and J36S plug types

Caution: Power cords and cables must not block access to Ethernet appliance components or drape where people can trip on them.

Note: In North America, AC power cords must not exceed 4.5 m (approximately 14.75 ft.) in length, to comply with National Electrical Code (NEC) Sections 400-8 (NFPA 75, 5-2.2) and 210-52 and Canadian Electrical Code (CEC) Section 4-010(3). The cords listed in Table 3-3 on page 75 are in compliance.

3.2.3 Power specifications and requirements for the IBM Ethernet Appliance J56S and J58S

The J56S Ethernet appliances are configurable with two, three, or four AC power supplies.

The J58S Ethernet appliances are configurable with three or four AC power supplies.

The power supplies connect to the mid plane, which distributes the different output voltages that are produced by the power supplies to the Ethernet appliance components, depending on their voltage requirements. Each power supply is cooled by its own internal cooling system.

Redundant power supplies are hot-removable and hot-insertable. When you remove a power supply from an Ethernet appliance that uses a non redundant power supply configuration, the Ethernet appliance may shut down depending on your configuration.

IBM Ethernet Appliance J56S AC Power Supply Description

Each AC power supply weighs approximately 5.0 lb. (2.3 kg) and consists of one AC appliance inlet, an AC switch, a fan, and LEDs to monitor the status of the power supply. Figure 3-4 shows the power supply.

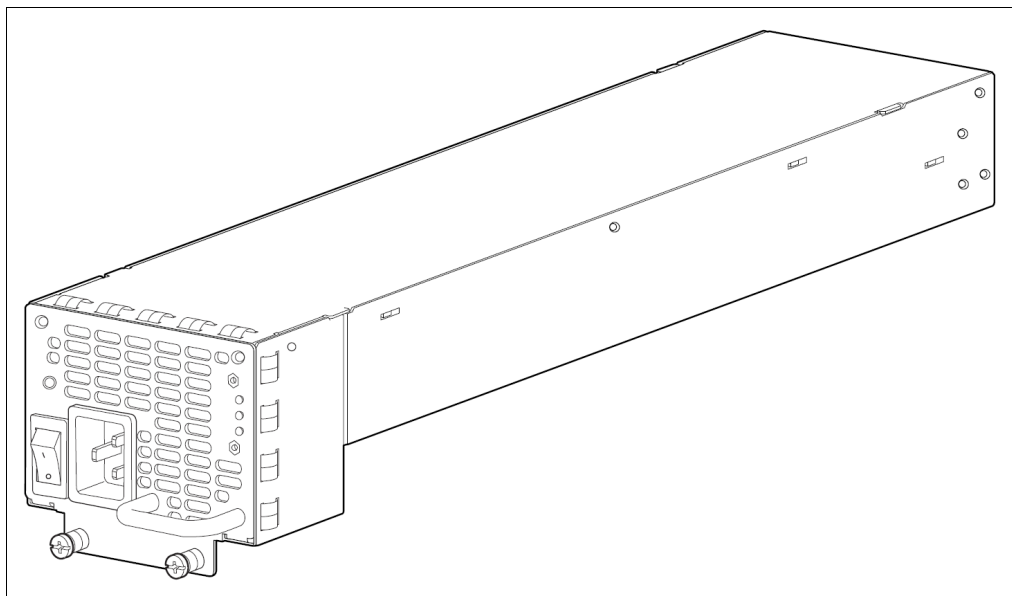


Figure 3-4 J56S power supply

Each inlet requires a dedicated AC power feed and a dedicated 15 A (250 VAC) circuit breaker.

J56S power supply LEDs

Each AC power supply faceplate contains three LEDs that indicate the status of the power supply. Table 3-4 provides information about the meaning of the various LED indicators.

Table 3-4 J56S AC power supply LEDs

Label	Color	State	Description
AC OK	Yellow	Off	AC power input voltage is below 78 VAC
	Green	On	AC power input voltage is within 78–264 VAC

Label	Color	State	Description
DC OK	Green	Off	DC power outputs generated by the power supply are not within the normal operating ranges
		On	DC power outputs generated by the power supply are within the normal operating ranges
PS FAIL	Red	Off	Power supply is functioning normally
		On	Power supply is not functioning normally and its output voltage is out of regulation limits. Check AC OK and DC OK LEDs for more information

AC power cord specifications for the J56S

Each AC power supply has a single AC appliance inlet located on the power supply that requires a dedicated AC power feed. Most sites distribute power through a main conduit that leads to frame-mounted power distribution panels, one of which can be located at the top of the rack that houses the router. An AC power cord connects each power supply to the power distribution panel.

Detachable AC power cords, each 2.5 m (approximately 8 ft.) long, are supplied with the appliance. The C19 appliance coupler at the female end of the cord inserts into the AC appliance inlet coupler, type C20 (right angle), as described by IEC standard 60320. The plug at the male end of the power cord fits into the power source receptacle that is standard for your geographical location.

Table 3-5 provides specifications and Figure 3-5 on page 79 depicts the plug on the AC power cord provided for each country or region.

Table 3-5 AC cord specifications for the J56S

Country or region	Electrical specifications	Plug standards
Australia	240 VAC, 50 Hz AC	SAA/3/15
China	220 VAC, 50 Hz AC	GH2-16P
Europe (except Denmark, Italy, Switzerland and United Kingdom)	220 or 230 VAC, 50 Hz AC	CEE 7/7
Italy	230 VAC, 50 Hz AC	CEI 23-16/VII
Japan	125 VAC 50 or 60 Hz AC 220 VAC, 50 or 60 Hz AC	JIS 8303 NEMA 6-20P
North America	125 VAC 60 Hz AC 125 VAC 60 Hz AC 250 VAC, 50 or 60 Hz AC 250 VAC, 50 or 60 Hz AC	NEMA 5-15P NEMA L5-15P NEMA 6-20 NEMA L6-20P
United Kingdom	240 VAC, 50 Hz AC	BS89/13

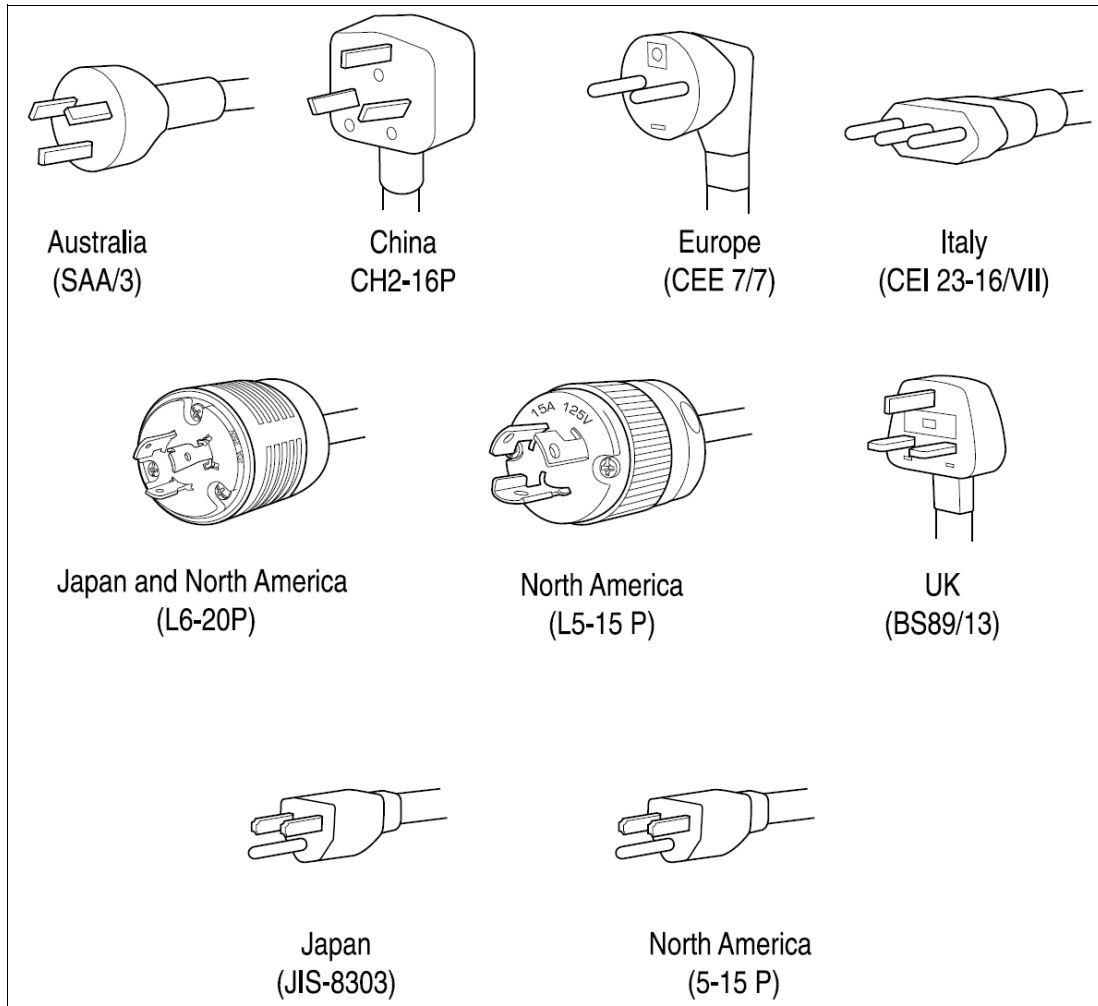


Figure 3-5 AC cord plug types for the J56S

IBM Ethernet Appliance J56S AC power supply description

The J58S Ethernet appliance contains three or four AC power supplies, shown in Figure 3-6 on page 80, located at the rear of the chassis in slots PEM0 through PEM3 (left to right).

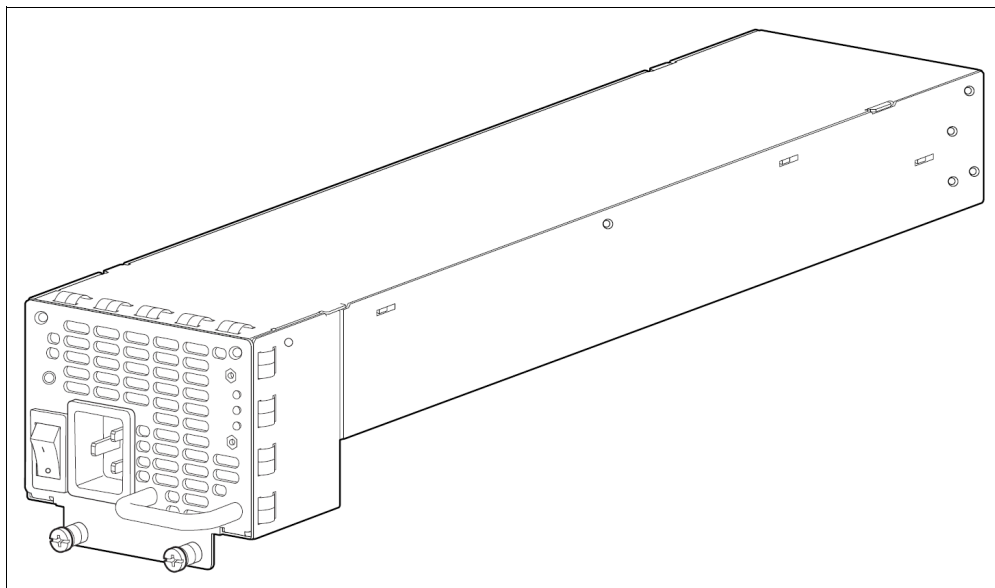


Figure 3-6 J56S power supply

Each AC power supply provides power to all components in the Ethernet appliance. When three power supplies are present, they share power almost equally within a fully populated system. Four AC power supplies provide full power redundancy. If one power supply fails or is removed, the remaining power supplies instantly assume the entire electrical load without interruption. Three power supplies provide the maximum configuration with full power for as long as the Ethernet appliance is operational. Each AC power supply has a corresponding AC appliance inlet located in the chassis directly above the power supply. Each inlet requires a dedicated AC power feed and a dedicated 15 A (250 VAC) circuit breaker.

J58S power supply LEDs

Each AC power supply faceplate contains three LEDs that indicate the status of the power supply, as shown in Table 3-6.

Table 3-6 J58S power supply LEDs

Label	Color	State	Description
AC OK	Green	Off	AC power applied to power supply is not within the normal operating range
		On	AC power applied to power supply is within the normal operating range
DC OK	Green	Off	DC power outputs generated by the power supply are not within the normal operating ranges
		On	DC power outputs generated by the power supply are within the normal operating ranges
PS FAIL	Red	Off	Power supply is functioning normally
		On	Power supply is not functioning normally and its output voltage is out of regulation limits. Check AC OK and DC OK LEDs for more information

Power supply notes:

- ▶ After a power supply is powered on, it can take up to 60 seconds for status indicators, such as the status LEDs on the power supply, and the chassis command display to indicate that the power supply is functioning normally. Ignore error indicators that appear during the first 60 seconds.
- ▶ If the system is completely powered off when you power on the power supply, the Routing Engine boots as the power supply completes its startup sequence. Normally, the services gateway boots from the Junos software on the Compact Flash card. After powering on a power supply, wait at least 60 seconds before turning it off.

AC power cord specifications for the J58S Appliance

Each AC power supply has a single AC appliance inlet located on the power supply that requires a dedicated AC power feed. Most sites distribute power through a main conduit that leads to frame-mounted power distribution panels, one of which can be located at the top of the rack that houses the router. An AC power cord connects each power supply to the power distribution panel.

Detachable AC power cords, each 2.5 m (approximately 8 ft.) long, are supplied with the router. The C19 appliance coupler at the female end of the cord inserts into the AC appliance inlet coupler, type C20 (right angle) as described by IEC standard 60320. The plug at the male end of the power cord fits into the power source receptacle that is standard for your geographical location.

Table 3-7 provides specifications for the J58S appliance.

Table 3-7 AC cord specifications for the J58S appliance

Country or region	Electrical specifications	Plug standards
Australia	240 VAC, 50 Hz AC	SAA/3
China	220 VAC, 50 Hz AC	PSB-10
Europe (except Denmark, Italy, Switzerland and United Kingdom)	220 or 230 VAC, 50 Hz AC	CEE 7/7
Italy	230 VAC, 50 Hz AC	CEI 23-16/VII
Japan	220 VAC, 50 or 60 Hz AC	NEMA 6-20P
North America	250 VAC, 50 or 60 Hz AC	NEMA L6-20P
United Kingdom	240 VAC, 50 Hz AC	BS89/13

Figure 3-7 on page 82 depicts the plug on the AC power cord provided for each country or region.

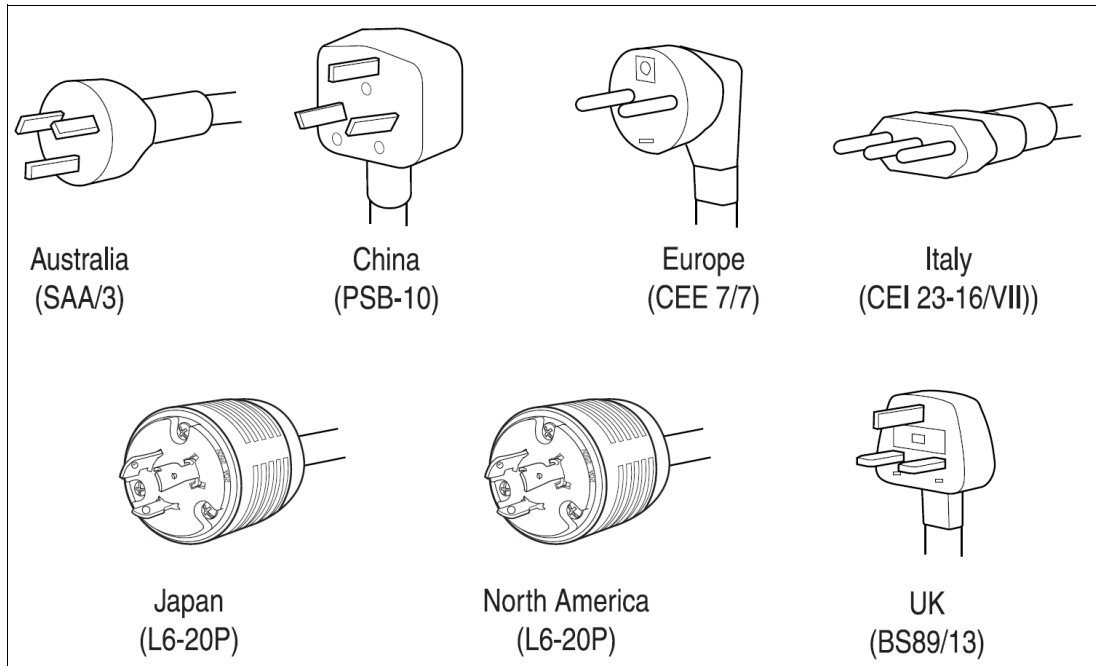


Figure 3-7 AC cord plug types for the J58S appliance

Caution: The AC power cord for an IBM e-series modular switch is intended for use with the switch only and not for any other use.

J58S chassis grounding specifications

To meet safety and electromagnetic interference (EMI) requirements and to ensure proper operation, the router must be adequately grounded before power is connected. To ground the appliance, connect a grounding cable to earth ground and then attach it to the chassis grounding points using the two screws provided. Two threaded inserts (PEM nuts) are provided on the right of the lower rear of the chassis for connecting the router to earth ground, as shown in Figure 3-8 on page 83.

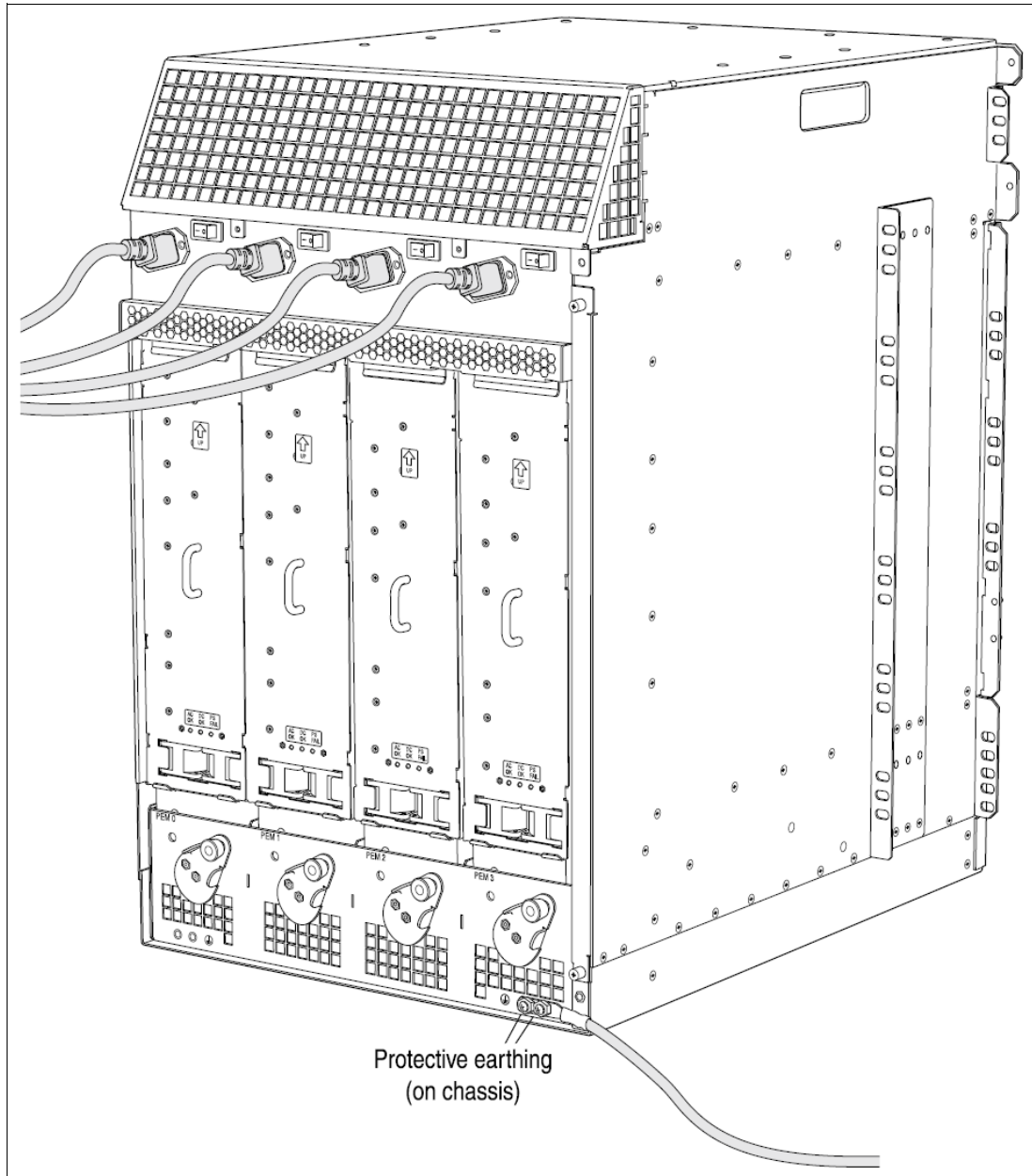


Figure 3-8 Grounding lug for the J58S

Danger: The J58S Ethernet Appliance is a pluggable type A equipment installed in a restricted-access location. It has a separate protective earthing terminal provided on the chassis in addition to the grounding pin of the power supply cord. This separate protective earthing terminal must be permanently connected to earth ground.

3.3 Cabling

In this section, we discuss details around cabling for the IBM j-type Ethernet Appliances. All IBM j-type Ethernet Appliances use the same cable types for the interfaces. Therefore, we do

not break this section into various equipment types; instead, we discuss cable types that apply to the IBM j-type Ethernet Appliances.

3.3.1 Cables connecting to management devices

There are two basic management ports: a console port and a management port. The management port might be labeled as an Ethernet port on some models of the IBM j-type equipment. However, the labeling difference does not affect the cable used for this out of band management port.

Console port

The cable used for the console port is an RS-232 (EIA-232) serial cable with an RJ45 on one end and a DB9 on the other end. The ports are configured as data terminal equipment (DTE). Table 3-8 shows the pinout for the DB9 end, and Table 3-9 shows the pinout for the RJ45 end.

Table 3-8 Console cable pinout for the DB9 connector

PIN	Signal	Description
1	DCD	Carrier Detect
2	RxD	Receive Data
3	TxD	Transmit Data
4	DTR	Data Terminal Ready
5	Ground	Signal Ground
6	DSR	Data Set Ready
7	RTS	Ready to Send
8	CTS	Clear to Send
9	Ring	Ring Indicator

Table 3-9 Console cable pinout for the RJ45 connector

PIN	Signal	Description
1	RTS	Request to Send
2	DTR	Data Terminal Ready
3	TxD	Transmit Data
4	Ground	Signal Ground
5	Ground	Signal Ground
6	RxD	Receive Data
7	DSR/DCD	Data Set Ready
8	CTS	Clear to Send

Note: The port labeled Aux has the same cable requirements as the console port.

Management port

In addition to the console and Aux ports, which require an RS-232 cable for connecting a console directly to the system, there is an Out-of-Band management port too. Depending on the j-type system in use, this port is labeled either Management (MGMT) or Ethernet. This port is an auto-sensing 10/100-Mbps Ethernet RJ-45 receptacle that accepts a standard Ethernet cable for connecting the system to a management LAN (or other device that supports out-of-band management). Table 3-10 describes the RJ-45 connector pinout.

Table 3-10 RJ-45 connector pinout for the Ethernet port

PIN	Signal
1	TX +
2	TX -
3	RX +
4	Termination Network
5	Termination Network
6	RX -
7	Termination Network
8	Termination Network

Standard RJ45 Ethernet ports

Excluding the Management ports used for Out-of-Band management, the IBM j-type systems provide RJ45 ports for Ethernet connectivity to LAN devices. These ports are auto-sensing and can accommodate several interface options. Fortunately, these interface options can all use the same cable. However, it is important to note that the pins carry different signals depending on what interface option is in use.

10/100 Ethernet cables

The pinout for 10/100 Ethernet is the same as the pinout listed in Table 3-10 for the Management port.

Power over Ethernet

While PoE uses the same Cat-5 cable as 10/100 Ethernet, it employs the unused lines to deliver power to an end device. Table 3-11 provides the pinout for PoE using an RJ45 connector.

Table 3-11 Pinout for PoE

PIN	Signal name	Description
1	TD +	Transmit Data
2	TD -	Transmit Data
3	RD +	Receive Data
4	Positive Vport	Power-Over-Ethernet
5	Positive Vport	Power-Over-Ethernet
6	RD -	Receive Data
7	Negative Vport	Power-Over-Ethernet

PIN	Signal name	Description
8	Negative Vport	Power-Over-Ethernet

Ethernet Bus 1000BaseT

In addition to 10/100 Ethernet and PoE, the standard RJ45 Ethernet ports also supports Gigabit Ethernet (GbE) using a Cat-5 cable. However, the pins of the cable are used in a different way than either of the other interfaces. The Ethernet 1000BaseT (Twisted Pair Pinout) is provided in Table 3-12.

Table 3-12 Ethernet Bus 1000BaseT RJ45 pinout

PIN	Signal name	Description
1	BI_DA+	Bi-directional pair +A
2	BI_DA-	Bi-directional pair -A
3	BI_DB+	Bi-directional pair +B
4	BI_DB-	Bi-directional pair -B
5	BI_DC+	Bi-directional pair +C
6	BI_DC-	Bi-directional pair -C
7	BI_DD+	Bi-directional pair +D
8	BI_DD-	Bi-directional pair -D

Optical ports

Throughout the IBM j-type Ethernet Appliances, optical ports are used for various types of connectivity, such as:

- ▶ 100Base-FX
- ▶ 1000Base-SX
- ▶ 1000Base-LX
- ▶ 1000Base-LH (or 1000Base-ZX)
- ▶ 10GBase-SR
- ▶ 10GBase-LR
- ▶ 10GBase-LRM
- ▶ Virtual Chassis Interconnect

All optical ports use standard small-form factor pluggable (SFP), or 10 gigabit small-form factor pluggable (XFP) transceivers, which require Lucent Connectors (LC).

Optical cables

Fiber-optic cables used with j-type equipment must match the specifications of the transceiver used for that particular port. Certain details, such as the mode (single mode or multi-mode) must be correct for the exact transceiver; alternately, other details, such as size (50 micron or 62.5 micron), must be sufficient for the particular connection, such as the distance that the signal must traverse.

Optical cable maintenance

Unlike copper cables, fiber optic cables require special attention and maintenance.

Note: The XFP cages and optics on the components are industry-standard parts that have limited tactile feedback for insertion of optics and fiber. You must insert the optics and fiber firmly until the latch is securely in place.

To maintain fiber-optic cables:

- ▶ When you unplug a fiber-optic cable from a transceiver, place rubber safety caps over the transceiver and on the end of the cable.
- ▶ Anchor fiber-optic cable to avoid stress on the connectors. When attaching a fiber-optic cable to a transceiver, be sure to secure the fiber-optic cable so that it is not supporting its own weight as it hangs to the floor. Never let a fiber-optic cable hang free from the connector.
- ▶ Avoid bending fiber-optic cables beyond their minimum bend radius. Bending fiber-optic cables into arcs smaller than a few inches in diameter can damage the cables and cause problems that are difficult to diagnose.
- ▶ Frequent plugging and unplugging of fiber-optic cables in and out of optical instruments can damage the instruments, which are expensive to repair. Attach a short fiber extension to the optical equipment. Any wear and tear due to frequent plugging and unplugging is then absorbed by the short fiber extension, which is easier and less expensive to replace than the instruments.
- ▶ Keep fiber-optic cable connections clean. Micro-deposits of oil and dust in the canal of the transceiver or cable connector can cause loss of light, reduction in signal power, and possibly intermittent problems with the optical connection.
- ▶ To clean the transceiver canal, use an appropriate fiber-cleaning device, such as Fiber Optic Adaptor Cleaning Wands (RIFOCS). Follow the directions in the cleaning kit you use. After cleaning the transceiver, make sure that the connector tip of the fiber-optic cable is clean. Use only an approved alcohol-free fiber-optic cable cleaning kit, such as the Optex Cletop-S Fiber Cleaner. Follow the directions in the cleaning kit that you use.

3.4 Cooling system

In the following sections, we discuss details around the IBM j-type Appliances cooling systems.

3.4.1 IBM Ethernet Appliance J34S cooling

The cooling system consists of a fan tray containing four fans and an air filter. These components work together to keep all Ethernet appliance components within the acceptable temperature range. The IBM Ethernet Appliance J34S has one fan tray located in the rear of the device that installs vertically to the right of the card cage. The fan tray is hot-insertable and hot-removable. Figure 3-9 on page 88 shows the fan tray.

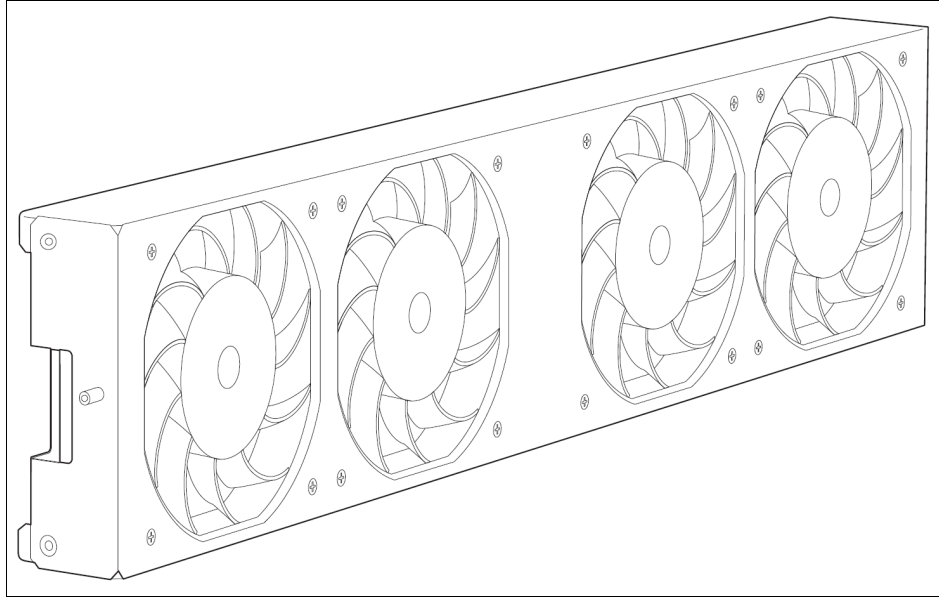


Figure 3-9 J34S fan tray

From the front of the chassis, there is a single air intake on the left side of the Ethernet appliance. Air is pushed from the fan tray through the air filter and then to the card cage. The air is exhausted out the right of the system. Figure 3-10 illustrates the air flow from the top of the chassis and the front of the chassis.

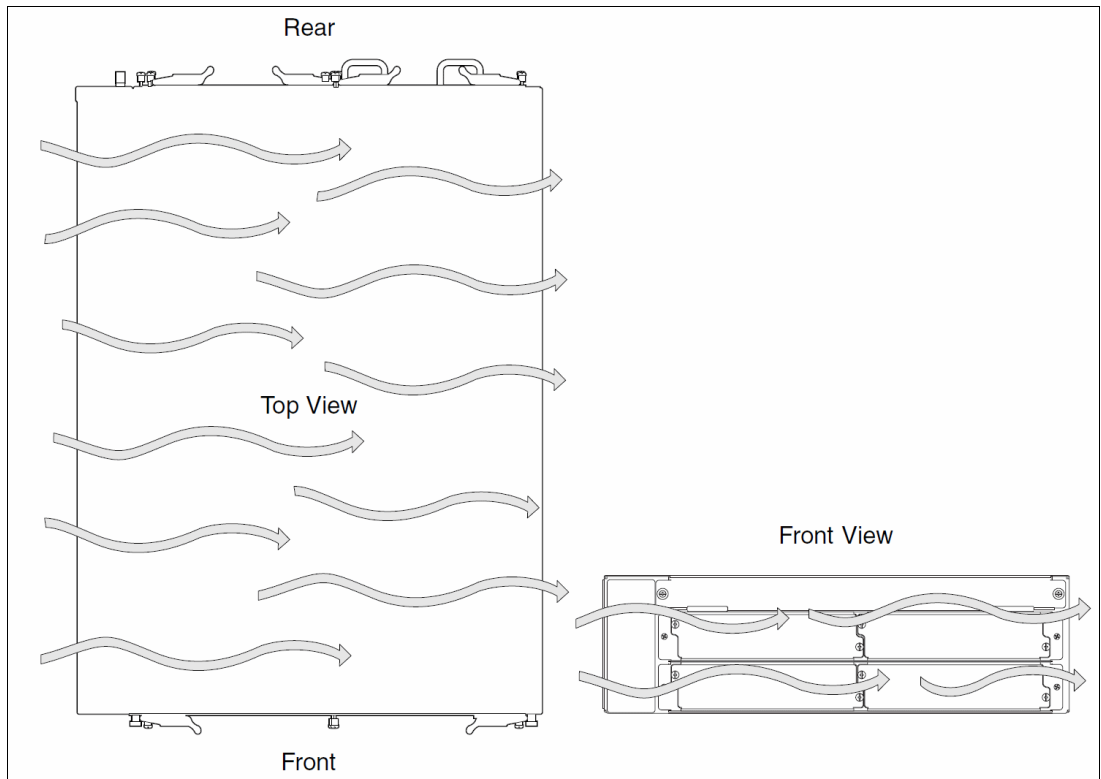


Figure 3-10 J34S air flow

The RE monitors the temperature of the device components. Above an ambient temperature of 30°C, the fans operate at full speed. Below 30°C, the fans operate at a reduced speed. If a

fan fails or the ambient temperature rises above a threshold, the speed of the remaining fans is automatically adjusted to keep the temperature within the acceptable range. If the ambient maximum temperature specification is exceeded and the system cannot be adequately cooled, the RE shuts down the system by disabling output power from each power supply.

Note: You must replace the fan tray within two minutes of removing it or the system shuts down.

3.4.2 IBM Ethernet Appliance J36S cooling

The cooling system consists of a fan tray containing 10 fans and an air filter. These components work together to keep all Ethernet appliance components within the acceptable temperature range. The IBM Ethernet Appliance J36S has one fan tray located in the rear of the device that installs vertically to the right of the card cage. The fan tray is hot-insertable and hot-removable. Figure 3-11 shows the fan tray.

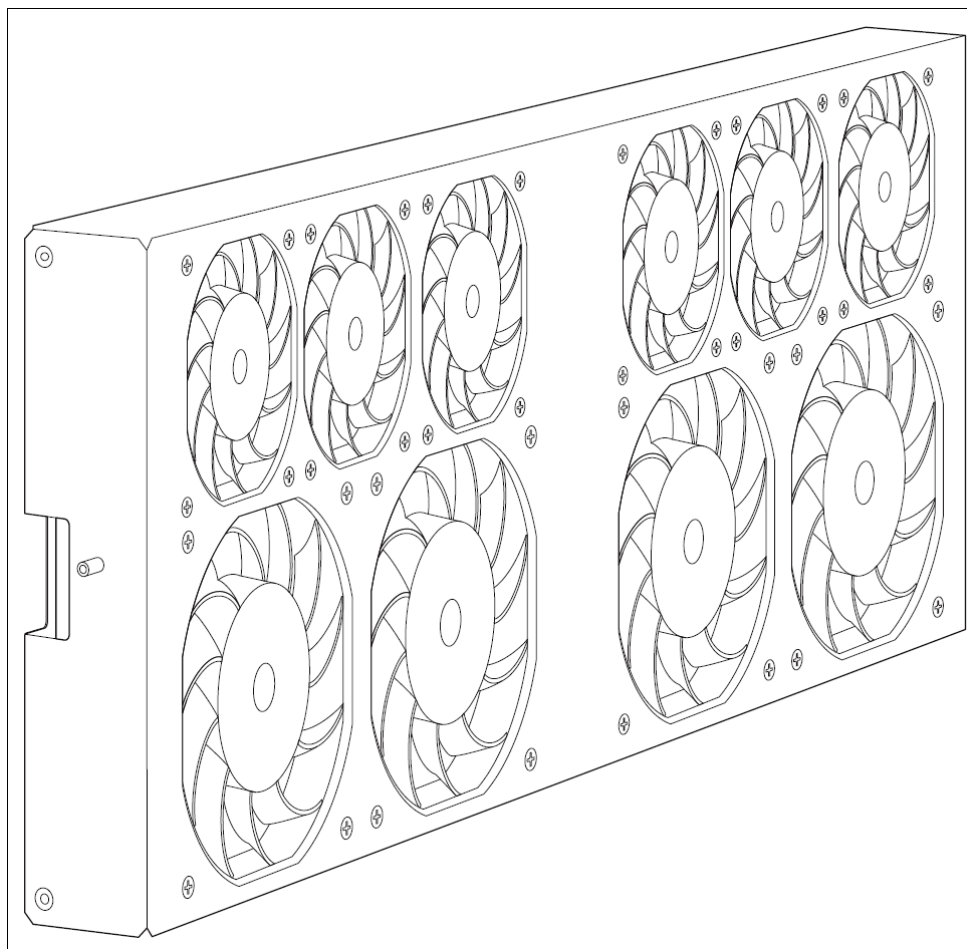


Figure 3-11 J36S fan tray

From the front of the chassis, there is a single air intake on the left side of the Ethernet appliance. Air is pushed from the fan tray through the air filter and then to the card cage. The air is exhausted out the right of the system. Figure 3-12 on page 90 illustrates the air flow from the top of the chassis and the front of the chassis.

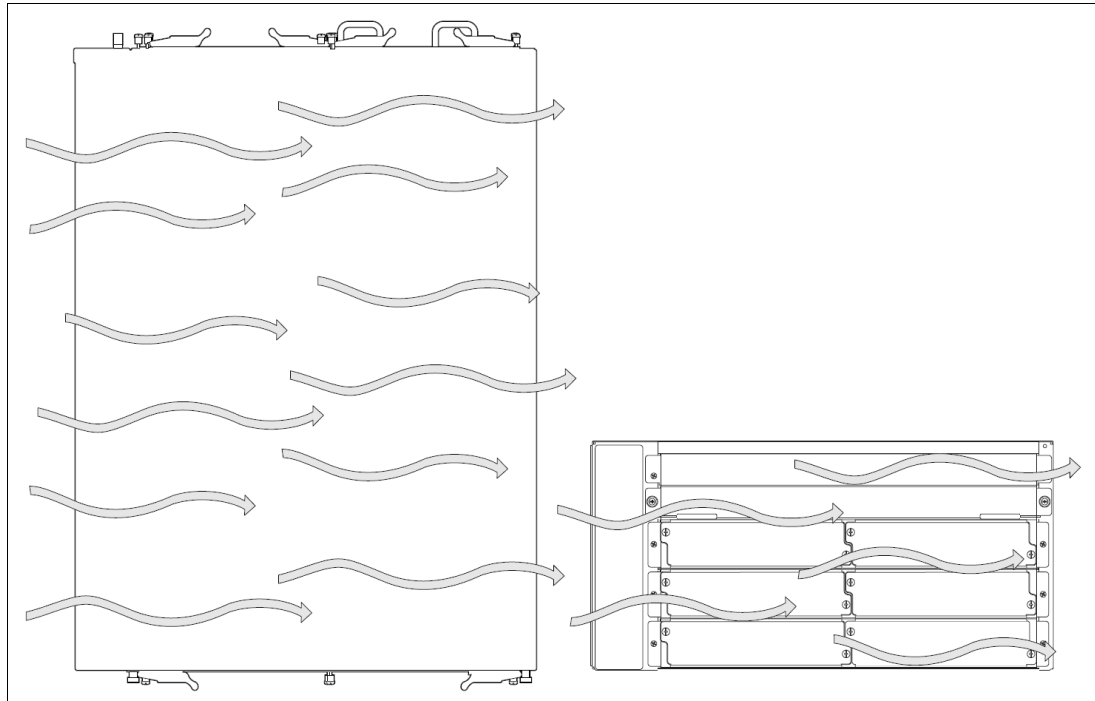


Figure 3-12 J36S air flow

The RE monitors the temperature of the device components. Above an ambient temperature of 30°C, the fans operate at full speed. Below 30°C, the fans operate at a reduced speed. If a fan fails or the ambient temperature rises above a threshold, the speed of the remaining fans is automatically adjusted to keep the temperature within the acceptable range. If the ambient maximum temperature specification is exceeded and the system cannot be adequately cooled, the RE shuts down the system by disabling output power from each power supply.

Note: You must replace the fan tray within two minutes of removing it or the system shuts down.

3.4.3 IBM Ethernet Appliance J56S cooling

The cooling system components work together to keep all appliance components within the acceptable temperature range. Figure 3-13 on page 91 shows the J56S fan tray and filter.

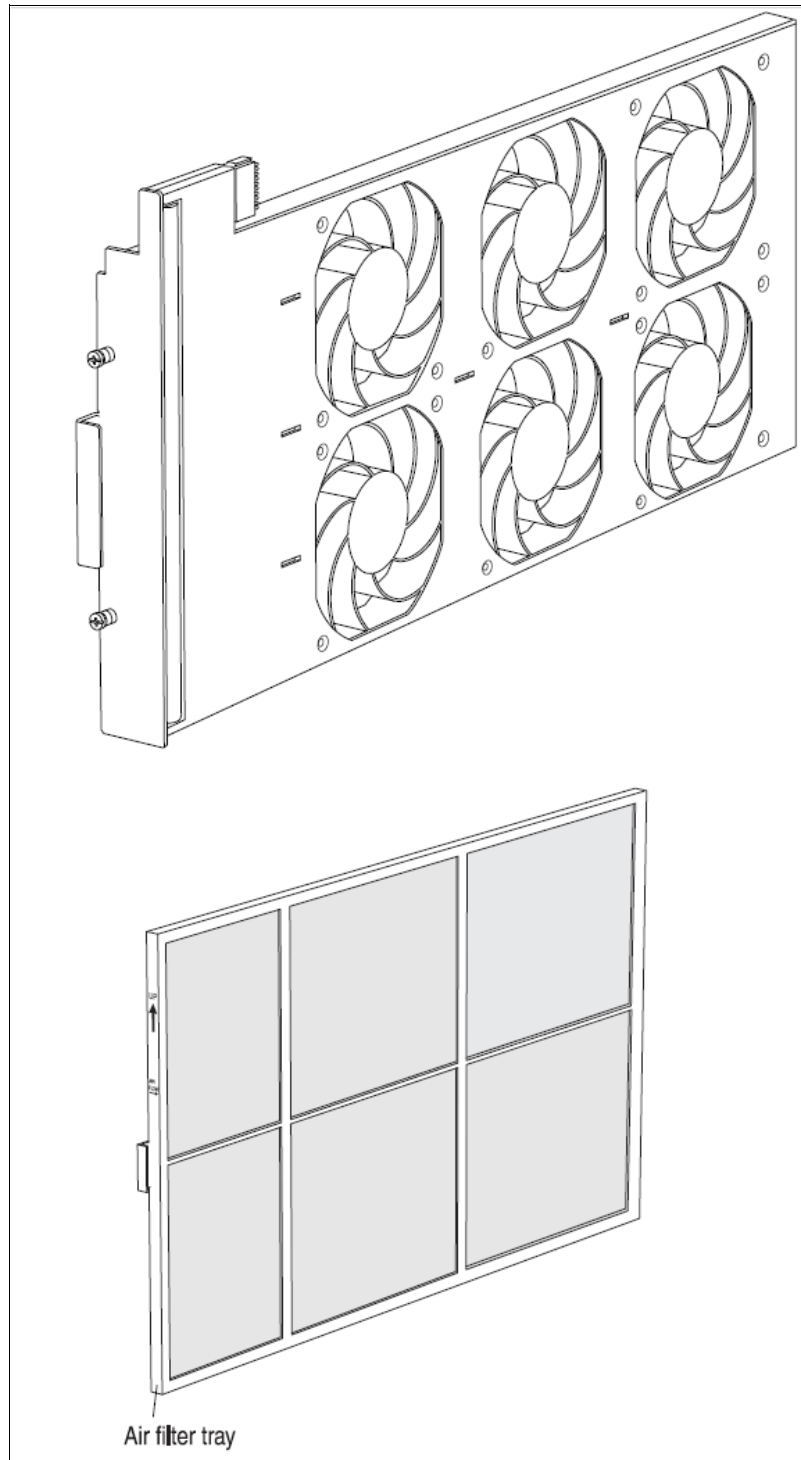


Figure 3-13 J56S fan tray and filter

The J56S has one fan tray and one air filter that installs vertically in the rear of the system. The fan tray contains six fans. The air intake to cool the chassis is located on the side of the chassis next to the air filter. Air is pulled through the chassis toward the fan tray, where it is exhausted out the side of the system. The air intake to cool the power supplies is located in the front of the router above the craft interface. The exhaust for the power supplies is located

on the rear bulkhead power supplies. Figure 3-14 on page 92 illustrates the air flow through the system.

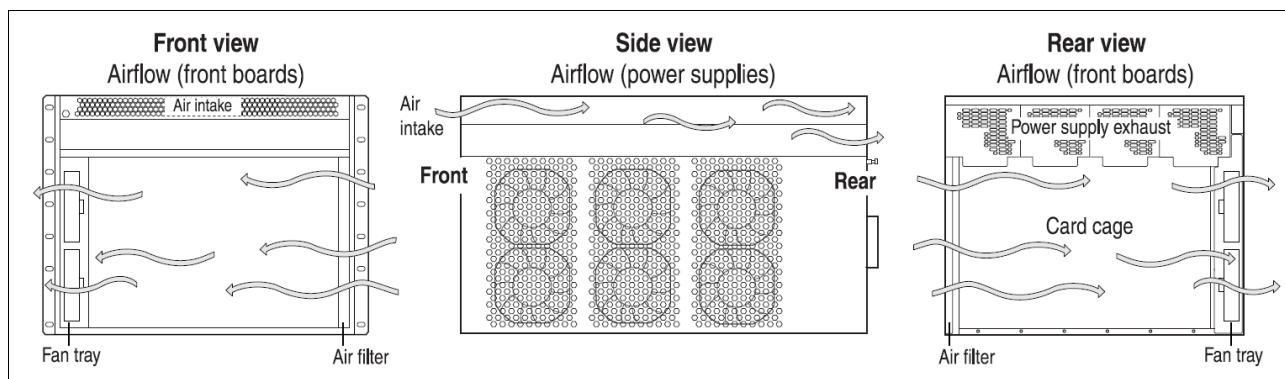


Figure 3-14 J56S air flow

The host subsystem monitors the temperature of the appliance components. When the system is operating normally, the fans function at lower than full speed. If a fan fails or the ambient temperature rises above a threshold, the speed of the remaining fans is automatically adjusted to keep the temperature within the acceptable range. If the ambient maximum temperature specification is exceeded and the system cannot be adequately cooled, the routing engine shuts down the system by disabling output power from each PEM.

Maintaining the J56S cooling system components

Regularly inspect the air filter. A dirty air filter restricts airflow in the unit, producing a negative effect on the ventilation of the chassis. The filter degrades over time. Periodically replace the filter in use and spares. We recommend that you replace the filter every six months.

Monitor the status of the fans. A fan tray contains multiple fans that work in unison to cool the router components. If one fan fails, the host subsystem adjusts the speed of the remaining fans to maintain proper cooling.

3.4.4 J58S cooling system description

The cooling system components work together to keep all appliance components within the acceptable temperature range. Figure 3-15 on page 93 shows several views of the J58S appliance's air flow.

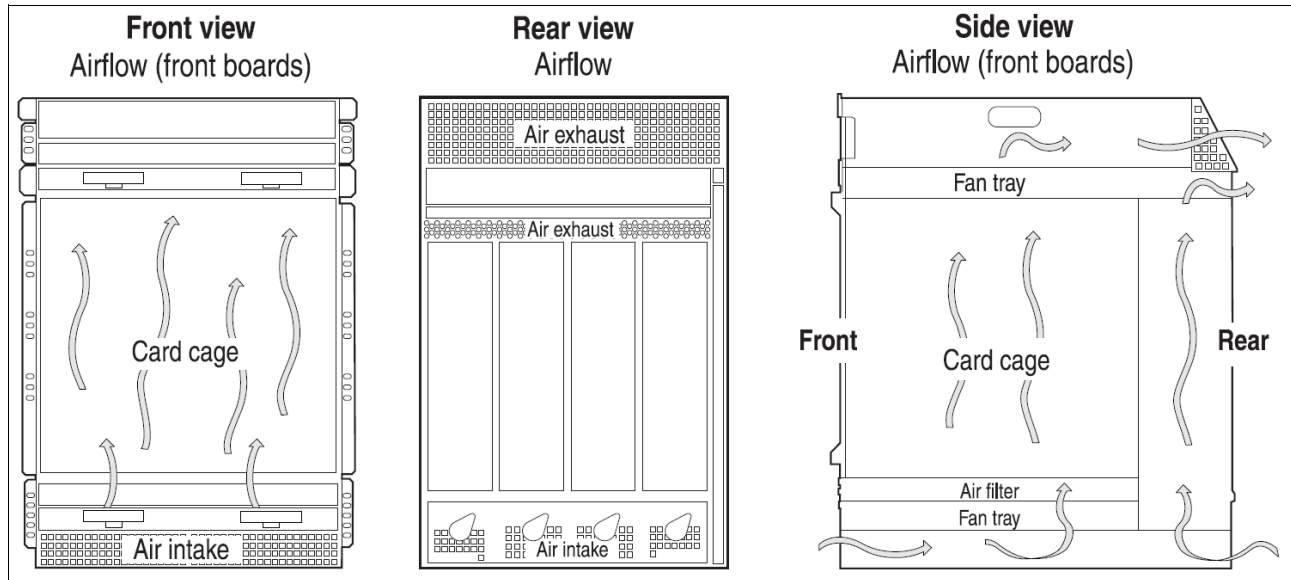


Figure 3-15 J58S air flow

The appliance has two fan trays located in the front of the system that install horizontally above and below the card cage. Each fan tray contains six fans. The fan trays are interchangeable and are hot-insertable and hot-removable. Figure 3-16 is an illustration of the fan tray.

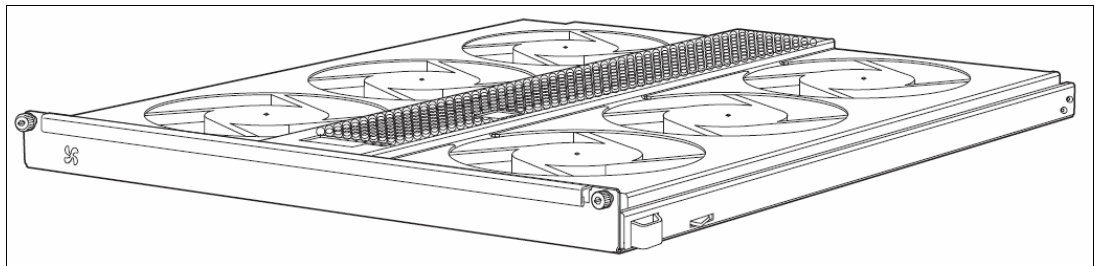


Figure 3-16 J58S fan tray

Additionally, as shown in Figure 3-17 on page 94, the J58S also has a pair of filters, and filter trays, that require occasional cleaning.

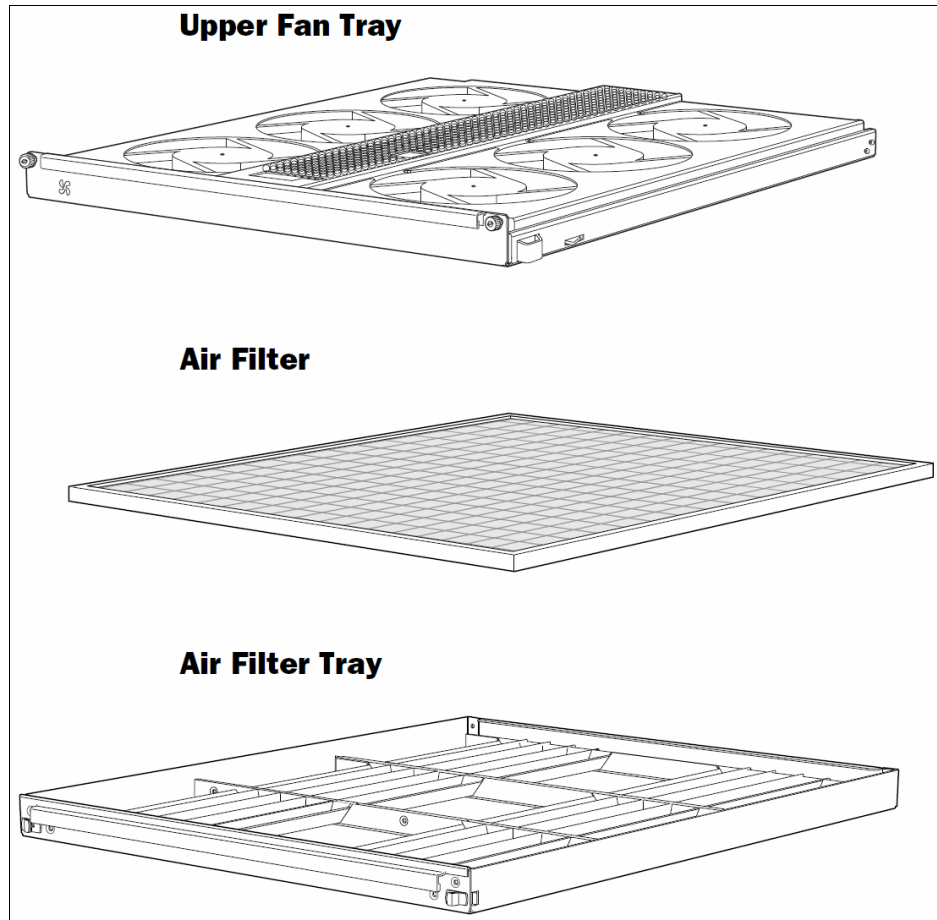


Figure 3-17 J58S fan tray and air filter

The host subsystem monitors the temperature of the system components. When the appliance is operating normally, the fans function at lower than full speed. If a fan fails or the ambient temperature rises above a threshold, the speed of the remaining fans is automatically adjusted to keep the temperature within the acceptable range. If the ambient maximum temperature specification is exceeded and the system cannot be adequately cooled, the routing engine shuts down the system by disabling output power from each PEM.

There is a single air intake in the front of the router. Air is pushed up through the card cage and through the upper fan tray where it combines in a common exhaust plenum and is exhausted out the upper rear of the system.

Maintaining the J58S cooling system components

Regularly inspect the air filter. A dirty air filter restricts airflow in the unit, producing a negative effect on the ventilation of the chassis. The filter degrades over time. Periodically replace the filter in use and spares. We recommend that you replace the filter every 6 months.

Monitor the status of the fans. A fan tray contains multiple fans that work in unison to cool the router components. If one fan fails, the host subsystem adjusts the speed of the remaining fans to maintain proper cooling. A red alarm is triggered when a fan fails, and a yellow alarm and red alarm is triggered when a fan tray is removed.

3.5 Racks

In the following sections, we discuss rack and cabinet mounting considerations for the IBM j-type Ethernet Appliances.

You can mount the IBM Ethernet Appliances in four-post open, or Telco racks. Two-post open rack installations are not supported. For additional information about the open rack requirements, refer to 3.6, “More information” on page 96, where numerous hardware guides are listed.

Additionally, the IBM Ethernet Appliances can be mounted in a cabinet that contains a 19-in. rack, as defined in *Cabinets, Racks, Panels, and Associated Equipment*, document number EIA-310-D, published by the Electronics Industry Association:

<http://www.eia.org>

3.5.1 Cabinet airflow requirements

When you mount the device in a cabinet, you must ensure that ventilation through the cabinet is sufficient to prevent overheating. Following is a list of requirements to consider when planning for chassis cooling:

- ▶ Ensure that the cool air supply that you provide through the cabinet can adequately dissipate the thermal output of the device.
- ▶ Ensure that the cabinet allows the chassis hot exhaust air to exit from the cabinet without recirculating into the device. An open cabinet (without a top or doors) that employs hot air exhaust extraction from the top allows the best airflow through the chassis. If the cabinet contains a top or doors, perforations in these elements assist with removing the hot air exhaust.
- ▶ Install the device as close as possible to the front of the cabinet so that the cable management system just clears the inside of the front door. This maximizes the clearance in the rear of the cabinet for critical airflow.
- ▶ Route and dress all cables to minimize the blockage of airflow to and from the chassis.

The minimum front and rear clearance requirements depend on the mounting configuration that you choose. The minimum total clearance inside the cabinet is 30.7 inches between the inside of the front door and the inside of the rear door.

IBM Ethernet Appliance J34S cabinet size and clearance requirements

The minimum size cabinet that can accommodate the device is 19 inches (482 mm) wide and 31.5 in. (800 mm) deep. A cabinet larger than the minimum requirement provides better airflow and reduces the chance of overheating. To accommodate a single device, the cabinet must be at least 13 U high. If you provide adequate cooling air and airflow clearance, you can stack eight devices in a cabinet that has at least 40 U (70 in. or 1.78 m) of usable vertical space.

IBM Ethernet Appliance J36S cabinet size and clearance requirements

The minimum size cabinet that can accommodate the device is 19 inches (482 mm) wide and 31.5 inches (800 mm) deep. A cabinet larger than the minimum requirement provides better airflow and reduces the chance of overheating. To accommodate a single device, the cabinet must be at least 13 U high. If you provide adequate cooling air and airflow clearance, you can stack eight devices in a cabinet that has at least 40 U (70 in. or 1.78 m) of usable vertical space.

IBM Ethernet Appliance J56S cabinet size and clearance requirements

The minimum size cabinet that can accommodate the device is 19 inches (482 mm) wide and 31.5 inches (800 mm) deep. A cabinet larger than the minimum requirement provides better airflow and reduces the chance of overheating. To accommodate a single device, the cabinet must be at least 13 U high. If you provide adequate cooling air and airflow clearance, you can stack five devices in a cabinet that has at least 48 U (84 in. or 2.13 m) of usable vertical space.

IBM Ethernet Appliance J58S cabinet size and clearance requirements

The minimum size cabinet that can accommodate the device is 23.62 inches (600 mm) wide and 31.5 inches (800 mm) deep. A cabinet larger than the minimum requirement provides better airflow and reduces the chance of overheating. To accommodate a single device, the cabinet must be at least 16 U high. If you provide adequate cooling air and airflow clearance, you can stack three devices in a cabinet that has at least 48 U (84 in. or 2.13 m) of usable vertical space.

3.5.2 Clearance for maintenance

For maintenance to access the unit:

- ▶ Leave at least 24 inches (61 cm) both in front of and behind the appliance. Allow at least six inches (15.2 cm) of clearance on each side of the chassis.
- ▶ Leave adequate space at the front of the appliances for service personnel to remove and install hardware components. NEBS GR-63 recommends that you allow at least 30 inches (76.2 cm) in front of the rack or cabinet and 24 inches (61 cm) behind the rack or cabinet.

3.6 More information

The installation of each model of the IBM j-type s-series systems are documented in both the Getting Started Guides and Hardware Guides for each system type.

For additional information and step-by-step installation procedures, refer to the appropriate documentation from the following list:

- ▶ *IBM Ethernet Appliance J34S Hardware Guide*, GA32-0748
- ▶ *IBM Ethernet Appliance J34S Getting Started Guide*, GA32-0749
- ▶ *IBM Ethernet Appliance J36S Hardware Guide*, GA32-0750
- ▶ *IBM Ethernet Appliance J36S Getting Started Guide*, GA32-0751
- ▶ *IBM Ethernet Appliance J56S Hardware Guide*, GA32-0752
- ▶ *IBM Ethernet Appliance J56S Getting Started Guide*, GA32-0753
- ▶ *IBM Ethernet Appliance J58S Hardware Guide*, GA32-0754
- ▶ *IBM Ethernet Appliance J58S Getting Started Guide*, GA32-0755

All of these documents are available on the IBM support site located at:

<http://www.ibm.com/systems/support/supportsite.wss/brandmain?brandind=5375876>

For more information about the Junos Software, refer to the following documentation:

- ▶ *Juniper Web Device Manager for IBM j-type Ethernet Switches and Routers Interface User Guide*, GA32-0688
- ▶ *JUNOS Software Access Privilege Configuration Guide*, GA32-0696
- ▶ *JUNOS Software Broadband Subscriber Management Solutions Guide*, GA32-0709

- ▶ *JUNOS Software Class of Service Configuration Guide*, GA32-0738
- ▶ *JUNOS Software CLI User Guide*, GA32-0697
- ▶ *JUNOS Software Configuration and Diagnostic Automation Guide*, GA32-0679
- ▶ *JUNOS Software Ethernet Routing Engine Media Upgrade Kit*, GA32-0681
- ▶ *JUNOS Software Feature Guide*, GA32-0739
- ▶ *JUNOS Software Hierarchy and RFC Reference*, GA32-0712
- ▶ *JUNOS Software High Availability Configuration Guide*, GA32-0670
- ▶ *JUNOS Software IBM j-type m-series Ethernet Routers Layer 2 Configuration Guide*, GA32-0708
- ▶ *JUNOS Software Installation and Upgrade Guide*, GA32-0695
- ▶ *JUNOS Software Interfaces Command Reference*, GA32-0672
- ▶ *JUNOS Software JUNOScript API Guide*, GA32-0674
- ▶ *JUNOS Software MPLS Applications Configuration Guide*, GA32-0702
- ▶ *JUNOS Software Multicast Protocols Configuration Guide*, GA32-0703
- ▶ *JUNOS Software NETCONF API Guide*, GA32-0678
- ▶ *JUNOS Software Network Interfaces Configuration Guide*, GA32-0706
- ▶ *JUNOS Software Network Management Configuration Guide*, GA32-0698
- ▶ *JUNOS Software Policy Framework Configuration Guide*, GA32-0704
- ▶ *JUNOS Software Routing Protocols and Policies Command Reference*, GA32-0673
- ▶ *JUNOS Software Services Interfaces Configuration Guide*, GA32-0707
- ▶ *JUNOS Software Subscriber Access Configuration Guide*, GA32-0711
- ▶ *JUNOS Software System Basics and Services Command Reference*, GA32-0671
- ▶ *JUNOS Software System Log Messages Reference*, GA32-0675
- ▶ *JUNOS Software VPNs Configuration Guide*, GA32-0705
- ▶ *JUNOScope Software User Guide*, GA32-0670



Junos fundamentals

In this chapter, we describe the main aspects of the Junos operating system. Each of the IBM j-type data center networking products runs a version of Junos, which is an operating system that was created by Juniper Networks and is built into the software architecture. It provides carrier-class network software to highly-available data centers of all sizes.

This chapter includes the following topics:

- ▶ Junos architecture
- ▶ Routing Engine
- ▶ Packet Forwarding Engine
- ▶ Junos software configuration model
- ▶ Junos software routing and forwarding tables
- ▶ User interface options
- ▶ Introduction to Junos CLI

4.1 Junos overview

Junos was designed based on the following key points, as depicted in Figure 4-1:

- One operating system

The most fundamental virtue of Junos software is a single source code base, which means that engineers can develop new features one time and then share the code, as applicable, across the many platforms running Junos OS.

A single, cohesive operating system that provides a consistent user experience makes planning easier, day-to-day operations more intuitive, and changes faster for customers. Administrators can configure and manage functionality from the basic chassis to complex routing using the same tools across devices to monitor, manage, and update the entire network.

- One software release train

The Junos approach to software development produces a stable code base that not only reduces the number of unplanned system events, but also the time and trouble of planned maintenance and upgrades.

New versions of Junos software are released in a defined quarterly schedule, providing stable and predictable delivery of new functionality.

- One modular software architecture

Each module of the Junos software runs in its own protected memory space, preventing one module from disrupting another. It also enables the independent restart of each module as necessary. This modular architecture provides for a high level of performance, high availability, security, and device scalability not found in other operating systems.

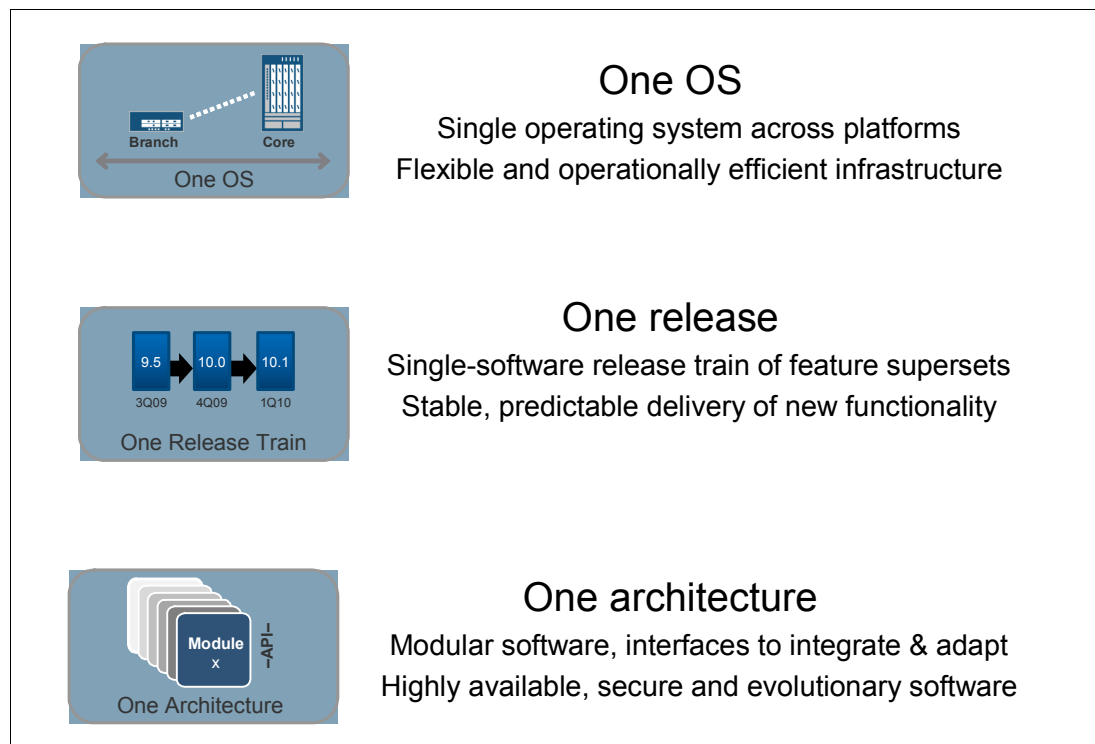


Figure 4-1 Junos overview

4.2 Junos architecture

Junos-based platforms share a common design that separates the switch's or router's control and forwarding planes, as shown in Figure 4-2. All IBM j-type family products consist of two major components:

- ▶ Routing Engine
- ▶ Packet Forwarding Engine

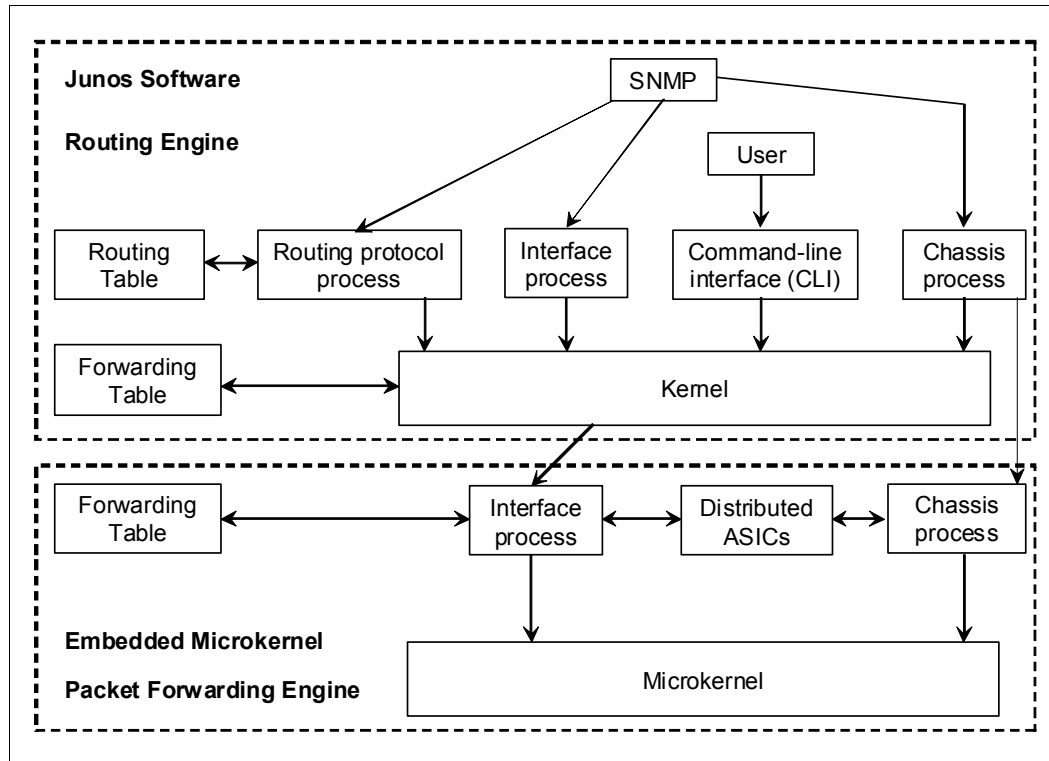


Figure 4-2 Junos architecture

4.3 Routing Engine

The Routing Engine controls the routing updates and system management. The Routing Engine consists of routing protocol software processes running inside a protected memory environment on a general-purpose computer platform. The Routing Engine handles all of the routing protocol processes and other software processes that control the routers' interfaces, some of the chassis components, system management, and user access to the router. These routers and software processes run on top of a kernel that interacts with the Packet Forwarding Engine.

4.3.1 Junos software Routing Engine components

The Junos software is based on the FreeBSD UNIX® operating system. The open source software was modified and hardened to operate in the router's specialized environment, for example, some executables were deleted and other utilities were de-emphasized. Additionally, certain daemons were added to enhance the routing functionality. The result of this transformation is the kernel, the heart of the Junos software. The kernel operates multiple daemons that perform the actual functions of the router. Each daemon operates in its own

protected memory space, which is also controlled by the kernel. This separation provides isolation between the processes and resiliency in the event of a process failure, which is important in a core routing platform because a single process failure does not cause the entire router to cease functioning.

The Junos software runs on the Routing Engine. As mentioned, the Junos software processes run on top of the kernel, which enables communication between processes and provides a direct link to the Packet Forwarding Engine software. The Junos software can be used to configure routing protocols and router interface properties and to monitor and troubleshoot protocol and network connectivity problems.

The following list describes some of the most important Junos software processes:

- **Routing Engine Kernel**

The Routing Engine kernel provides the underlying infrastructure for all Junos Software processes. In addition, it provides the link between the routing tables and the Routing Engine's forwarding table. It is also responsible for all communication with the Packet Forwarding Engine.

- **Initialization process**

An initialization process starts and monitors all the other software processes when the router boots. If a software process terminates or fails to start, the initialization process attempts to restart it a limited number of times and logs any failure information for further investigation.

- **Management process**

The management process manages the configuration of the router and all user commands. The management process is responsible for notifying other daemons when a new configuration is committed.

- **Routing protocol process**

The routing protocol process controls the routing protocols that run on the router. This process starts all configured routing protocols and handles all routing messages. It maintains one or more routing tables, which consolidate the routing information learned from all routing protocols. From this routing information, the routing protocol process determines the active routes to network destinations and installs these routes into the Routing Engine's forwarding table. Finally, it implements routing policy, which enables you to control the routing information that is transferred between the routing protocols and the routing table. You can also filter routing information using routing policy and set properties associated with routes.

- **Interface process**

Using the interface process you can configure and control the physical interface devices and logical interfaces that are present in a device. You can configure interface properties, such as the interface location, the interface encapsulation, and interface-specific properties. You can configure the interfaces currently present in the router, as well as interfaces that are not present but that you might add later. The Junos interface process communicates through the kernel with the interface process in the Packet Forwarding Engine, enabling it to track the status and condition of the router's interfaces.

- **Chassis process**

Using the chassis process you can configure and control the properties of the router, including conditions that trigger alarms. The chassis process on the Routing Engine communicates directly with its peer processes running on the Packet Forwarding Engine.

- **SNMP process**

The Junos software supports the Simple Network Management Protocol (SNMP), which helps administrators monitor the state of a router. The SNMP software is controlled by the Junos SNMP and Management Information Base II processes, which consist of an SNMP master agent and various subagents.

4.4 Packet Forwarding Engine

The Packet Forwarding Engine (PFE) forwards packets through the device. The PFE is implemented using specific Application-Specific Integrated Circuits (ASICs). The separation between control operations, such as protocol updates and system management from packet forwarding provides superior performance and reliable deterministic operation.

The PFE receives the forwarding table and bridging table from the Routing Engine through an internal link. Forwarding table and bridging table updates are a high priority for the software kernel.

The PFE forwards packets based on the Layer 2 and Layer 3 forwarding tables that are provided by the RE. In addition, it also implements a number of advanced services. Some examples of advanced services are policers that provide rate limiting, firewall filters, and class of service.

4.5 Junos software configuration model

The router configuration is saved using a commit model: a candidate configuration is modified as desired and then committed to the system. The process is shown in Figure 4-3

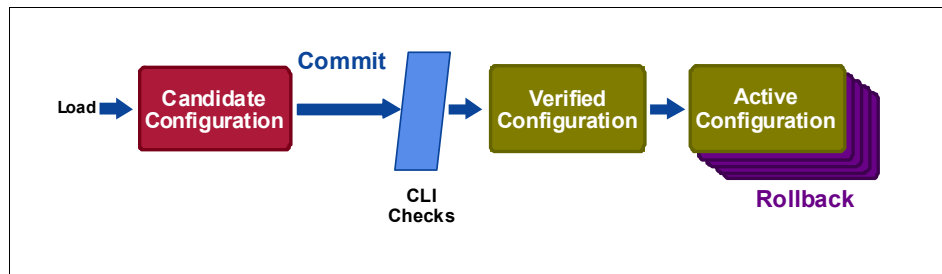


Figure 4-3 Commit model

4.5.1 Active configuration

Configuration changes to the Junos software do not take effect immediately. The active configuration is the configuration that is currently operational in the device.

4.5.2 Candidate configuration

The candidate configuration is a temporary configuration that might become the active configuration. Initially, Junos creates a candidate configuration and populates it with the switch's active configuration. You can modify this configuration and commit the changes, which causes the candidate configuration to become the active configuration.

4.5.3 Configuration history

After it is committed:

- ▶ The router checks the configuration for syntax errors, and if no errors are found, the configuration is saved as `juniper.conf.gz` and activated.
- ▶ The former active configuration file is saved as the first rollback configuration file (`juniper.conf.1.gz`)
- ▶ All other rollback configuration files are incremented by one, for example, `juniper.conf.1.gz` is incremented to `juniper.conf.2.gz`, making it the second rollback configuration file.
- ▶ The router can have a maximum of 49 rollback configurations (1–49) saved on the system.

On the router, the active configuration file and the first three rollback files (`juniper.conf.gz.1`, `juniper.conf.gz.2`, `juniper.conf.gz.3`) are located in the `/config` directory. If the recommended rescue file `rescue.conf.gz` is saved on the system, this file must also be saved in the `/config` directory. The factory default files are located in the `/etc/config` directory.

There are two mechanisms used to propagate the configurations between Routing Engines within a router:

- ▶ Synchronization

Propagates a configuration from one Routing Engine to a second Routing Engine within the same router chassis. To synchronize a router's configurations, use the **`commit synchronize`** CLI command. If one of the Routing Engines is locked, the synchronization fails. If synchronization fails because of a locked configuration file, you can use the `commit synchronize force` command that overrides the lock and synchronizes the configuration files.

- ▶ Distribution

Propagates a configuration across the routing plane on a multichassis router. Distribution occurs automatically. There is no user command available to control the distribution process. If a configuration is locked during a distribution of a configuration, the locked configuration does not receive the distributed configuration file, so the synchronization fails. You must clear the lock before the configuration and resynchronize the routing planes.

4.6 Junos routing and forwarding tables

A major function of the Junos routing protocol process is to maintain the Routing Engine's routing tables and from these tables determine the active routes to network destinations. The routing protocol process then installs these routes into the Routing Engine's forwarding table. The Junos kernel then copies this forwarding table to the Packet Forwarding Engine. The routing protocol process maintains multiple routing tables. By default, it maintains the following three routing tables. You can configure additional routing tables to suit your requirements.

- ▶ Unicast routing table

Stores routing information for all unicast routing protocols running on the router. BGP, IS-IS, OSPF, and RIP all store their routing information in this routing table. You can configure additional routes, such as static routes, to be included in this routing table. BGP, IS-IS, OSPF, and RIP use the routes in this routing table when advertising routing information to their neighbors.

- Multicast routing table

Stores routing information for all the running multicast protocols. DVMRP and PIM both store their routing information in this routing table, and you can configure additional routes to be included in this routing table

- MPLS routing table

Stores MPLS path and label information

With each routing table, the routing protocol process uses the collected routing information to determine active routes to network destinations. For unicast routes, the routing protocol process determines active routes by choosing the most preferred route, which is the route with the lowest preference value. By default, the route's preference value is simply a function of how the routing protocol process learned about the route. You can modify the default preference value using routing policy and with software configuration parameters. For multicast traffic, the routing protocol process determines active routes based on traffic flow and other parameters specified by the multicast routing protocol algorithms. The routing protocol process then installs one or more active routes to each network destination into the Routing Engine's forwarding table.

4.7 User interface options

You can use any of the following methods to configure Junos Software:

- Junos Command-Line Interface (CLI)

The Junos CLI is a straightforward command interface. You use Emacs-style keyboard sequences to move around on a command line and scroll through a buffer that contains recently executed commands. You type commands on a single line, and the commands are executed when you press the Enter key. The CLI also provides command help and command completion

- ASCII File

You can load an ASCII file containing a router configuration that you created earlier, either on this system or another system. You can then activate and run the configuration file as is, or you can edit it using the CLI and then activate it.

- J-Web

As an alternative to entering CLI commands, the Junos Software supports a J-Web graphical user interface (GUI). The J-Web user interface enables you to monitor, configure, troubleshoot, and manage the router on a client by means of a Web browser with Hypertext Transfer Protocol (HTTP) or HTTP over Secure Sockets Layer (HTTPS) enabled.

- JUNOScript API

The JUNOScript API is an Extensible Markup Language (XML) application that client applications use to request and change configuration information about IBM j-type routers. This API is customized for Junos software, and operations in the API are equivalent to Junos CLI configuration mode commands. The JUNOScript API includes a set of Perl modules that enable client applications to communicate with a JUNOScript server on the router. The Perl modules are used to develop custom applications for configuring and monitoring Junos Software.

► **NETCONF API**

The NETCONF API is an Extensible Markup Language (XML) application that client applications can use to request and change configuration information about IBM j-type routers. This API is customized for Junos software, and includes features that accommodate the configuration data models of multiple vendors. The NETCONF API includes a set of Perl modules that enable client applications to communicate with a NETCONF server on the router. The Perl modules are used to develop custom applications for configuring and monitoring Junos Software.

► **Configuration commit scripts**

You can create and use scripts that run at commit time to enforce custom configuration rules. If a configuration breaks the custom rules, the script can generate actions that the Junos software performs. These actions include:

- Generating custom error messages
- Generating custom warning messages
- Generating custom system log messages
- Making changes to the configuration

Configuration commit scripts also enable you to create macros, which expand simplified custom aliases for frequently used configuration statements into standard Junos configuration statements. Commit scripts are written in Extensible StylesheetLanguage Transformations (XSLT).

4.8 Introduction to Junos CLI

Junos CLI is a specific command shell that runs on top of a UNIX-based operating system kernel. The CLI provides command help and command completion.

The CLI also provides a variety of UNIX utilities, such as Emacs-style keyboard sequences, that allow you to move around on a command line and scroll through recently executed commands, regular expression matching to locate and replace values and identifiers in a configuration, filter command output, or log file entries, store and archive router files on a UNIX-based file system, and exit from the CLI environment and create a UNIX C shell or Bourne shell to navigate the file system, manage switch processes, and so on.

The CLI has two modes, operational mode and configuration mode.

4.8.1 Operational mode

In operational mode, you enter commands to monitor and troubleshoot switch hardware and software and network connectivity. Operational mode is indicated by the > prompt, for example:

```
user@host>
```

For more information about operational mode, refer to Chapter 6, “User interface” on page 119.

4.8.2 Configuration mode

In configuration mode, you can define all properties of the Junos software, including interfaces, routing protocols, user access, security features, and several system hardware properties.

To enter configuration mode, type the **configure** command:

```
user@host> configure
```

In configuration mode, you are actually viewing and changing the candidate configuration file. The candidate configuration allows you to make configuration changes without causing operational changes to the current operating configuration, called the active configuration. When you commit the changes you added to the candidate configuration, the system updates the active configuration. Using candidate configurations you can alter your configuration without causing potential damage to your current network operations.

For more information about configuration mode, refer to Chapter 6, “User interface” on page 119.

4.8.3 More information

For more information about Junos software and basic configuration, refer to the *System Basics Configuration Guide*:

http://www.juniper.net/techpubs/en_US/junos10.0/information-products/topic-collections/config-guide-system-basics/config-guide-system-basics.pdf

For more information about Junos CLI refer to *CLI User Guide*:

http://www.juniper.net/techpubs/en_US/junos10.1/information-products/topic-collections/swconfig-cli/swconfig-cli.pdf



Initial configuration

This chapter covers the following topics:

- ▶ Initial appliance configuration using the Junos software
- ▶ Configuring Junos software for the first time on an appliance with a Single Routing Engine
- ▶ Initial Junos software configuration on an appliance with Dual Routing Engines
- ▶ Junos software default settings for appliance security
- ▶ Junos software configuration using the CLI
- ▶ Activating the Junos Software Candidate Configuration
- ▶ Disk space management for Junos software installation

5.1 Initial IBM j-type Appliance configuration

This topic provides an overview of initial appliance configuration tasks using the Junos software.

On most Junos-based appliances, the Junos software is installed on the CompactFlash card and on the hard disk. When you first turn on an appliance, it runs the version of the Junos software installed on the CompactFlash card. The copy of Junos software that is on the hard disk is a backup. Another backup copy of the Junos software is available on removable media, such as a PC Card or a CompactFlash card. Be sure to put the backup Junos software that is on the removable media in a safe place.

When you turn on an appliance for the first time, the Junos software automatically boots and starts. You must enter basic configuration information so that the appliance can connect to the network, and you can log in to it.

To configure the appliance initially, you must connect a terminal or mobile computer to the appliance through the console port, which is a serial port on the front of the appliance. Only console access to the appliance is enabled by default. Remote management access to the appliance and all management access protocols, including Telnet, FTP, and SSH, are disabled by default.

When you first connect to the appliance console, you must log in as the user root. At first, the root account requires no password. You see that you are the user root because the appliance command prompt shows the username root@#.

You must start the Junos software command-line interface (CLI) using the command `cli`. The command prompt root@> indicates that you are the user root and that you are in the Junos software operational mode. Enter the Junos software configuration mode by typing the command **configure**. The command prompt root@# indicates that you are in the Junos software configuration mode.

When you first configure an appliance, you must configure the following basic properties:

- ▶ The appliance host name
- ▶ The domain name
- ▶ The IP address of the appliance Ethernet management interface. On all appliances the management Ethernet interface is fxp0.
- ▶ The IP address of a backup appliance
- ▶ The IP address of one or more DNS name servers on your network
- ▶ The Password for the root account

5.1.1 Configuring the Junos software for the first time on an appliance with a Single Routing Engine

When you turn on an appliance the first time, the Junos software automatically boots and starts. You must enter basic configuration information so that the appliance is in the network and you can log in to it over the network.

To configure the appliance initially, you must connect a terminal or mobile computer to the appliance through the console port, a serial port on the front of the appliance. Only console access to the appliance is enabled by default. Remote management access to the appliance

and all management access protocols, including Telnet, FTP, and SSH, are disabled by default.

You must gather the following information before configuring the appliance:

- ▶ Name the router uses in the network
- ▶ Domain name the router uses
- ▶ IP address and prefix length information for the Ethernet interface
- ▶ IP address of a default router
- ▶ IP address of a DNS server
- ▶ Password for the root user

To configure the Junos software for the first time on a appliance with a single Routing Engine:

1. Connect a terminal or mobile computer to the appliance through the console port, which is a serial port on the front of the appliance. Only console access to the appliance is enabled by default.
2. Power on the appliance, and wait for it to boot. The Junos software boots automatically. The boot process is complete when you see the login prompt on the console.
3. Open a terminal emulator application (such as HyperTerminal or putty.exe on a Windows workstation or TERM in a UNIX environment), and configure as follows:
 - a. In a Microsoft Windows environment, adjust the following parameters and values if necessary:
 - Bits per second: 9600
 - Databits: 8
 - Parity: None
 - Stop bits: 1
 - Flow control: None

Figure 5-1 shows the putty serial connection configuration options.

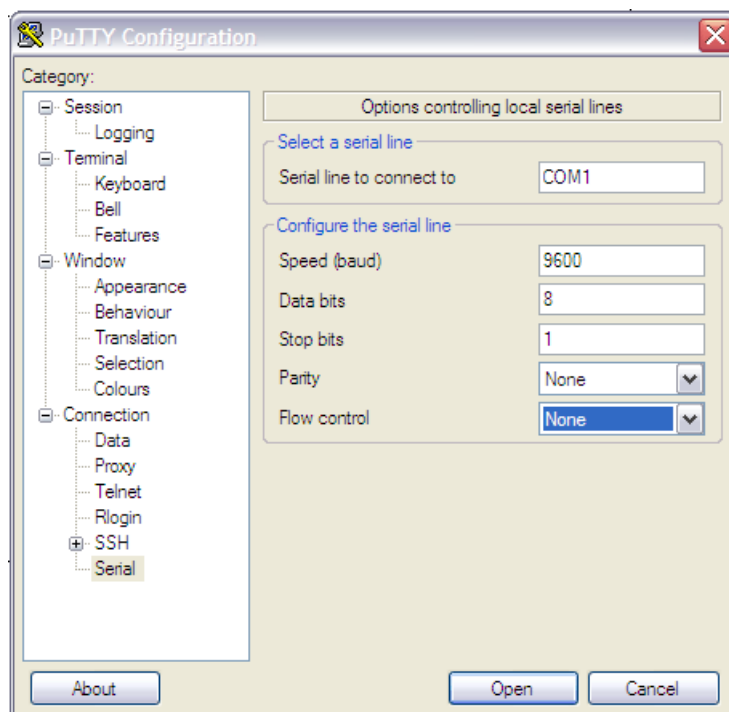


Figure 5-1 Serial port settings

- b. In a UNIX environment, enter the following string at the prompt:

```
tip /dev/ttyb -9600
```

4. Log in as the user root. Initially, the root user account requires no password. You can see that you are the root user because the prompt on the appliance shows the username root@#.

5. Start the Junos software command-line interface:

```
root@# cli
root@>
```

6. Enter Junos software configuration mode:

```
root@> configure
[edit]
root@#
```

7. Configure the name of the appliance (the appliance host name). We do not recommend spaces in the appliance name. However, if the name does include spaces, enclose the entire name in quotation marks (" ").

```
[edit]
root@# set system host-name host name
```

8. Configure the appliance's domain name:

```
[edit]
root@# set system domain-name domain-name
```

9. Configure the IP address and prefix length for the appliance management Ethernet interface. The management Ethernet interface provides a separate out-of-band management network for the appliance.

```
[edit]
root@# set interfaces fxp0 unit 0 family inet address address/prefix-length
```

10. Configure the IP address of a backup or default router. This device is called the backup router because it is used only while the routing protocol process is not running. Choose a router that is directly connected to the local appliance by way of the management interface. The appliance uses this backup router only when it is booting and only or when the Junos routing software, the routing protocol process, or rpd is not running.

For appliances with two Routing Engines, the backup Routing Engine, RE1, uses the backup router as a default gateway after the appliance boots. This enables you to access the backup Routing Engine. RE0 is the default master Routing Engine.

```
[edit]
root@# set system backup-router address
```

11. Configure the IP address of a DNS server. The router uses the DNS name server to translate host names into IP addresses.

```
[edit]
root@# set system name-server address
```

12. Set the root password, entering either a clear-text password that the system encrypts, a password that is already encrypted, or an SSH public key string.

Choose one of the following:

- a. To enter a clear-text password, use the following command:

```
[edit]
root@# set system root-authentication plain-text-password
New password: type password
Retype new password: retry password
```

- b. To enter a password that is already encrypted, use the following command:

```
[edit]
root@# set system root-authentication encrypted-password
encrypted-password
```

- c. To enter an SSH public key, use the following command:

```
[edit]
root@# set system root-authentication ssh-rsa key
```

13. Optionally, display the configuration statements, as shown in Example 5-1.

Example 5-1 Display configuration

```
[edit]
root@ show
system {
  host-name hostname;
  domain-name domain.name;
  backup-router address;
  root-authentication {
    [encrypted-password "password" | public-key];
    ssh-rsa "public-key";
    ssh-dsa "public-key";
  }
  name-server {
    address;
  }
  interfaces {
    fxp0 {
      unit 0 {
        family inet {
          address address ;
        }
      }
    }
  }
}
```

14. Commit the configuration, which activates the configuration on the appliance:

```
[edit]
root@# commit
```

After committing the configuration, you see the newly configured hostname appear after the username in the prompt—for example, **user@host#**.

Junos Software defaults are now set on the router.

If you want to configure additional Junos Software properties at this time, remain in the CLI configuration mode and add the necessary configuration statements. You need to commit your configuration changes to activate them on the appliance.

15. Exit from the CLI configuration mode.

```
[edit]
root@hostname# exit
root@hostname>
```

16. Back up the configuration on the hard drive.

After you have installed the software on the appliance, committed the configuration, and are satisfied that the new configuration is successfully running, you must issue the **request system snapshot** command to back up the new software to the /altconfig file system. If you do not issue the **request system snapshot** command, the configuration on the alternate boot device will be out of sync with the configuration on the primary boot device.

The **request system snapshot** command causes the root file system to be backed up to /altroot, and /config to be backed up to /altconfig. The root and /config file systems are on the appliance's CompactFlash card, and the /altroot and /altconfig file systems are on the router's hard disk.

Note: After you issue the request system snapshot command, you cannot return to the previous version of the software because the running copy and the backup copy of the software are identical.

5.1.2 Configuring the Junos software for the first time on an appliance with Dual Routing Engines

If a router has dual Routing Engines, you must initially configure each router independently. The sequence is irrelevant.

Configure the host names and addresses of the two Routing Engines using configuration groups at the **[edit groups]** hierarchy level. Use the reserved configuration group **re0** for the Routing Engine in slot 0 and **re1** for the Routing Engine in slot 1 to define properties specific to the individual Routing Engines. Configuring **re0** and **re1** groups enables both Routing Engines to use the same configuration file.

Use the **apply-groups** statement to reproduce the configuration group information in the main part of the configuration.

The **commit synchronize** command commits the same configuration on both Routing Engines. The command makes the active or applied configuration the same for both Routing Engines with the exception of the groups, **re0** being applied to only **RE0** and **re1** being applied only to **RE1**. If you do not synchronize the configurations between two Routing Engines and one of them fails, the router might not forward traffic correctly because the backup Routing Engine might have a different configuration.

To initially configure a router with dual Routing Engines, follow these steps:

1. Go to 5.1.1, “Configuring the Junos software for the first time on an appliance with a Single Routing Engine” on page 110, and follow Step 1 through Step 5 to initially configure the backup Routing Engine.
2. Instead of Step 6 and Step 8 in “Configuring the Junos software for the first time on an appliance with a Single Routing Engine” on page 110, configure a host name for each Routing Engine and an IP address for each management Ethernet interface fxp0, as follows:

```
[edit]
root@# edit groups
[edit groups]
root@# set re0 system host-name router1
root@# set re0 interfaces fxp0 unit 0 family inet address 10.10.10.1/24
root@# set re1 system host-name router2
root@# set re1 interfaces fxp0 unit 0 family inet address 10.10.10.2/24
```

3. Configure the appliance’s domain name:

```
[edit]
root@# set system domain-name domain-name
```

4. Set the loopback interface address for each Routing Engine.

```
[edit groups]
root@# set re0 interfaces lo0 unit 0 family inet address 2.2.2.1/32
root@# set re1 interfaces lo0 unit 0 family inet address 2.2.2.2/32
```

5. Configure the apply-groups statement to reproduce the configuration group information to the main part of the configuration.

```
[edit groups]
root@# top
[edit]
root@# set apply-groups [re0 | re1]
```

6. Configure Routing Engine redundancy:

```
[edit]
root@# set chassis redundancy routing-engine 0 master
root@# set chassis redundancy routing-engine 1 backup
root@# set chassis redundancy routing-engine graceful-switchover
```

7. Save the configuration change on both Routing Engines:

```
[edit]
user@host> commit synchronize
root@#
```

8. Continue with Step 9 through Step 12 in “Configuring the Junos software for the first time on an appliance with a Single Routing Engine” on page 110.

9. After you install the new software and are satisfied that it is successfully running, issue the **request system snapshot** command to back up the new software on both master and backup Routing Engines:

```
{master}
```

```
user@host> request system snapshot
```

The root file system is backed up to /altroot, and /config is backed up to /altconfig. The root and /config file systems are on the appliance's CompactFlash card, and the /altroot and /altconfig file systems are on the appliance's hard disk.

Note: After you issue the request system snapshot command, you cannot return to the previous version of the software, because the running copy and backup copy of the software are identical.

5.1.3 Junos software default settings for appliance security

The Junos software protects against common appliances security weaknesses with the following default settings:

- ▶ The Junos software does not forward directed broadcast messages. Directed broadcast services send ping requests from a spoofed source address to a broadcast address and can be used to attack other Internet users. Here is an example: if broadcast ping messages were allowed on the 200.0.0.0/24 network, a single ping request can result in up to 254 responses to the supposed source of the ping. The source might actually become the victim of a denial-of-service (DoS) attack.
- ▶ Only console access to the appliance is enabled by default. Remote management access to the appliance and all management access protocols, including Telnet, FTP, and SSH, are disabled by default.
- ▶ The Junos software does not support the SNMP **set** capability for editing configuration data. Although the software supports the SNMP **set** capability for monitoring and troubleshooting the network, this support exposes no known security issues. You can configure the software to disable this SNMP **set** capability.
- ▶ The Junos software ignores martian addresses that contain the following prefixes: 0.0.0.0/8, 127.0.0.0/8, 128.0.0.0/16, 191.255.0.0/16, 192.0.0.0/24, 223.255.55.0/24, and 240.0.0.0/4. Martian addresses are reserved host or network addresses about which all routing information must be ignored.

5.1.4 Junos software configuration using the CLI

You configure the Junos software using the Junos Command Line Interface (CLI). The CLI is described in detail in the *JUNOS Software CLI User Guide GA32-0697-00* available from

<http://www-947.ibm.com/systems/support/supportsite.wss/brandmain?brandind=5375876>

After completing the initial minimal configuration, you can configure software properties. If you configure the software interactively using the CLI, you enter software configuration statements to create a candidate configuration that contains a hierarchy of statements. At any hierarchy level, you generally can enter statements in any order. While you are configuring the software, you can display all or portions of the candidate configuration, and you can insert or delete statements. Any changes you make affect only the candidate configuration, not the active configuration that is running on the appliance.

The configuration hierarchy logically groups related functions, which results in configuration statements that have a regular, consistent syntax. For example, you configure routing protocols, routing policies, interfaces, and SNMP management in their own separate portions of the configuration hierarchy.

At each level of the hierarchy, you can display a list of the statements available at that level, along with short descriptions of the statements' functions. To have the CLI complete the statement name if it is unambiguous or to provide a list of possible completions, you can type a partial statement name followed by a space or tab.

More than one user can edit a appliance's configuration simultaneously. All changes made by all users are visible to everyone editing the configuration.

5.1.5 Activation of the Junos software candidate configuration

You enter software configuration statements using the CLI to create a candidate configuration that contains a hierarchy of statements. To have a candidate configuration take effect, you commit the changes. At this point, the candidate file is checked for proper syntax, activated, and marked as the current, operational software configuration file. If multiple users are editing the configuration, when you commit the candidate configuration, all changes made by all the users take effect.

The CLI always maintains a copy of previously committed versions of the software configuration. If you need to return to a previous configuration, you can do this from within the CLI.

5.1.6 Disk space management for Junos software installation

A Junos software installation or upgrade might fail if your appliance has a shortage of disk space. If a disk space error occurs, use one or more of the following options to complete the installation:

- ▶ Use the **request system storage cleanup** command to delete unnecessary files and increase storage space on the router.
- ▶ Specify the **unlink** option when you use the **request system software add** command to install the Junos software:

Download the software packages you need from the Juniper Networks Support Web site:

<http://www.juniper.net/support/partners/ibm/>

The download program provides intelligent disk space management to enable installation.

5.1.7 Junos software tools for monitoring the router

The primary method of monitoring and troubleshooting the Junos software, routing protocols, network connectivity, and the router hardware is to enter commands from the CLI. The CLI enables you to display information in the routing tables and routing protocol-specific data and to check network connectivity using **ping** and **traceroute** commands.

The J-Web graphical user interface (GUI) is a web-based alternative to using CLI commands to monitor, troubleshoot, and manage the appliance. See Figure 5-2 on page 118 for a view of the monitor page from a J-Web session.

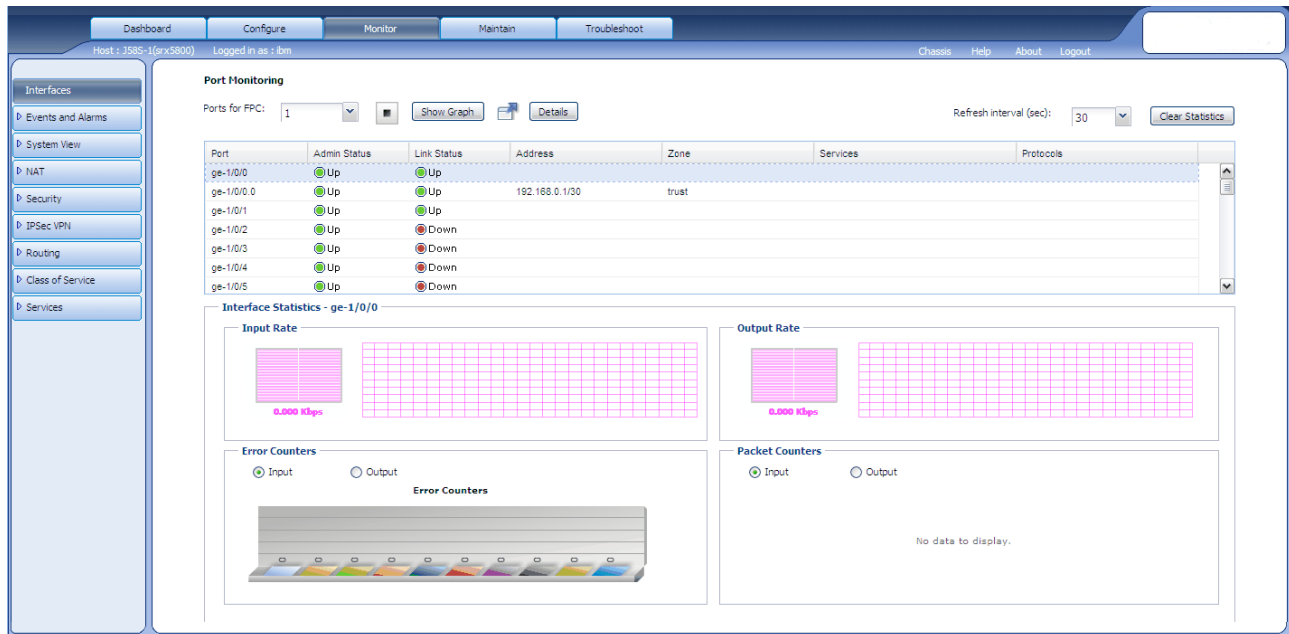


Figure 5-2 J-Web Monitor Page

The Junos software includes SNMP software, which enables you to manage routers. The SNMP software consists of an SNMP master agent and a MIB II agent, and supports MIB II SNMP version 1 traps and version 2 notifications, SNMP version 1 **Get** and **GetNext** requests, and version 2 **GetBulk** requests.

The software also supports tracing and logging operations so that you can track events that occur in the appliance, both normal appliance operations and error conditions, and track the packets that are generated by or pass through the appliance. Logging operations use a syslog-like mechanism to record system-wide, high-level operations, such as interfaces going up or down and users logging in to or out of the appliance. Tracing operations record more detailed messages about the operation of routing protocols, such as the various types of routing protocol packets sent and received, and routing policy actions.

5.2 More information

For additional information and step-by-step installation procedures, refer to the appropriate documentation from the following list.

- ▶ *IBM Ethernet Appliance J34S Hardware Guide*, GA32-0748
- ▶ *IBM Ethernet Appliance J34S Getting Started Guide*, GA32-0749
- ▶ *IBM Ethernet Appliance J36S Hardware Guide*, GA32-0750
- ▶ *IBM Ethernet Appliance J36S Getting Started Guide*, GA32-0751
- ▶ *IBM Ethernet Appliance J56S Hardware Guide*, GA32-0752
- ▶ *IBM Ethernet Appliance J56S Getting Started Guide*, GA32-0753
- ▶ *IBM Ethernet Appliance J58S Hardware Guide*, GA32-0754
- ▶ *IBM Ethernet Appliance J58S Getting Started Guide*, GA32-0755

All these documents are available on the IBM support site located at:

<http://www.ibm.com/systems/support/supportsite.wss/brandmain?brandind=5375876>



User interface

In this chapter, we discuss the user interfaces that you can use to configure, manage, monitor, and troubleshoot IBM j-type s-series Ethernet Appliance. They are:

- ▶ J-Web graphical user interface (GUI)
- ▶ Junos command line interface (CLI)

While the most of the configuration steps in this publication are done are using CLI, we provide a brief introduction for the J-Web GUI. For more information about the J-Web GUI, refer to the *JUNOS Software J-Web Interface User Guide Release 10.1* at:

http://jnpr.net/techpubs/en_US/junos10.1/information-products/topic-collections/jweb-user-guide/frameset.html

6.1 J-Web graphical user interface

J-Web is a web-based graphical user interface (GUI) that you can access by using either Hypertext Transfer Protocol (HTTP) or HTTP Secure (HTTPS). It provides a user-friendly GUI to perform some basic configuration tasks and complex setting using CLI Tools from J-Web GUI.

J-Web GUI is installed by default on IBM j-type s-series Ethernet Appliance. You can use J-Web GUI for initial configuration.

6.1.1 Logging into J-Web GUI

Before you can actually access the J-Web, you must ensure that HTTP or HTTPS is enabled. You must also require a valid login account to log in as you are authenticated, as shown in Figure 6-1 on page 121.

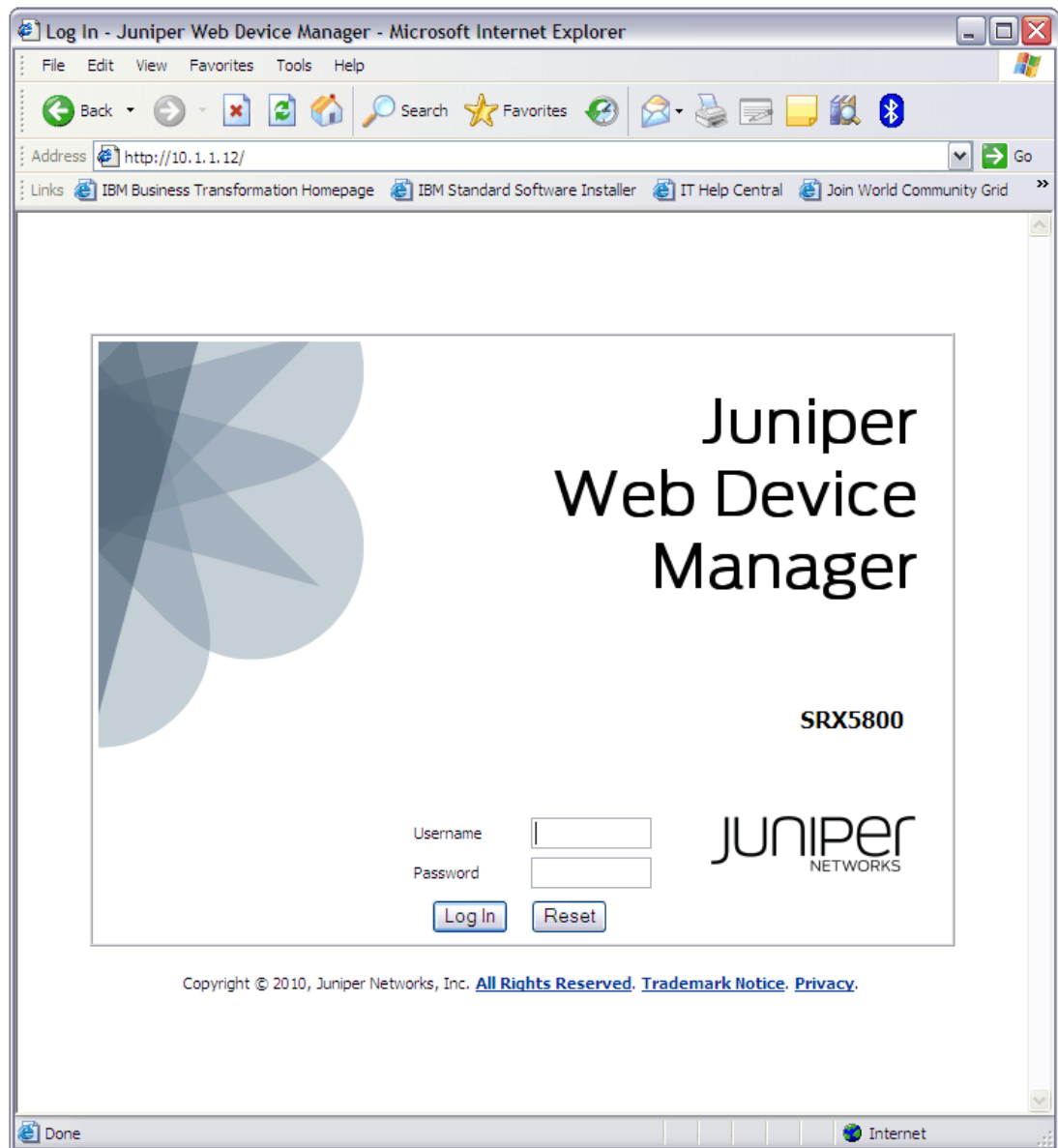


Figure 6-1 J-Web login page

Use Internet Explorer version 6.0 and higher, or Firefox version 2.0 and higher to access the J-Web interface.

Secure web access overview

A routing platform uses the Secure Sockets Layer (SSL) protocol to provide secure management of routing platforms through the Web interface. SSL uses public-private key technology that requires a paired private key and an authentication certificate for the SSL service. SSL encrypts communication between your routing platform and the Web browser with a session key negotiated by the SSL server certificate.

An SSL certificate includes identifying information such as a public key and a signature made by a certificate authority (CA). When you access the routing platform through HTTPS, an SSL handshake authenticates the server and the client and begins a secure session. If the information does not match or the certificate has expired, you are not able to access the routing platform through HTTPS.

Without SSL encryption, communication between your routing platform and the browser is sent in the open and can be intercepted. We recommend that you enable HTTPS access on your WAN interfaces.

On j-type switches, and routers, HTTP access is enabled by default on the built-in management interfaces. By default, HTTPS access is supported on any interface with an SSL server certificate.

Generating SSL certificates

To enable secure Web access, you must first generate a digital SSL certificate, and then enable HTTPS access on the routing platform.

To generate an SSL certificate:

1. Enter the following **openssl** command in your Secure Shell (SSH) command-line interface. The **openssl** command generates a self-signed SSL certificate in the privacy-enhanced mail (PEM) format. It writes the certificate and an unencrypted 1024-bit RSA private key to the specified file:

```
% openssl req -x509 -nodes -newkey rsa:1024 -keyout filename.pem -out filename.pem
```

2. Replace the filename with the name of a file in which you want the SSL certificate to be written—for example, new.pem.
3. When prompted, type the appropriate information in the identification form, for example, type US for the country name.
4. Display the contents of the file new.pem:

```
cat new.pem
```

5. Copy the contents of this file for installing the SSL certificate.

To install the SSL certificate and enable HTTPS, follow the procedure as outlined in “Configuring secure web access” on page 122.

Configuring secure web access

To configure secure web access:



1. Navigate to the Management Access Configuration page by selecting **Configuration**  **System Properties**  **Management Access**.
2. On the main page, click Edit. The Edit Management Access page is displayed. On this page, you can enable HTTP and HTTPS access on interfaces for managing Services Routers through the Web interface. You can also install SSL certificates and enable JUNOScript over SSL with the Secure Access page.

Figure 6-2 on page 123 shows the Edit Management Access page.

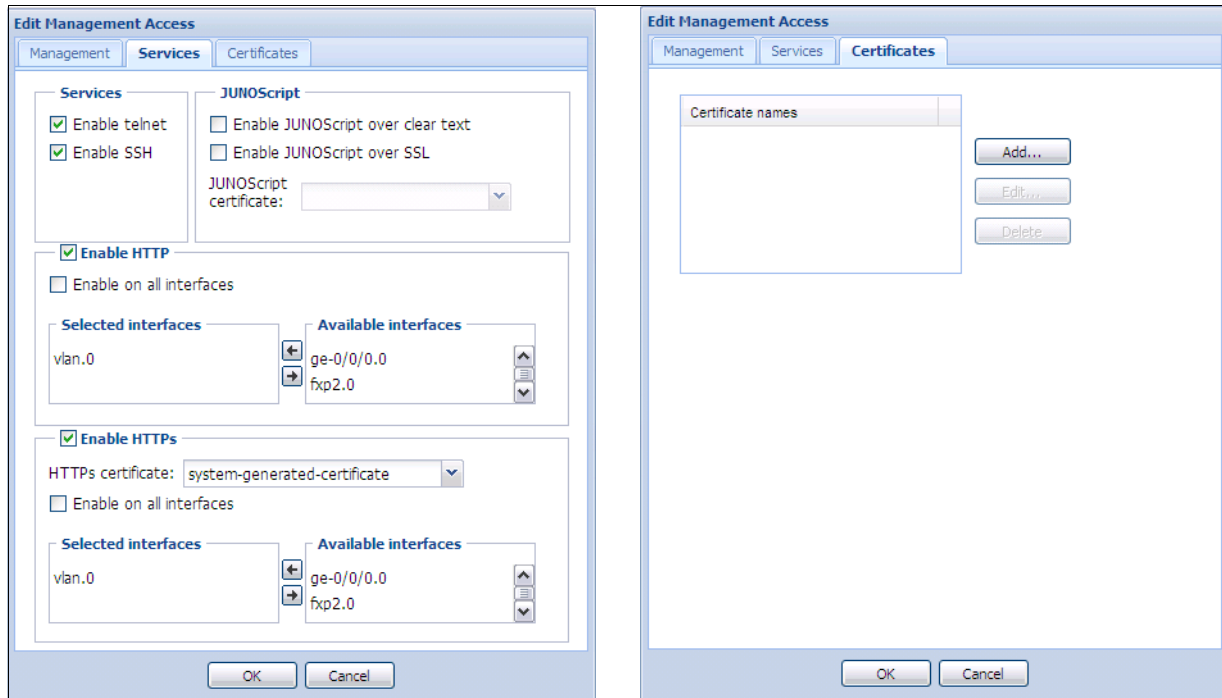


Figure 6-2 Edit Management Access page

To configure Web access settings in the J-Web interface, enter the information into the Edit Management Access page as described in the following lists:

► HTTP Web Access:

- Enable HTTP Access: To enable HTTP access, select the **Enable HTTP access** option on the Services tab.
- Enable HTTP on All Interfaces: To enable HTTP access on all interfaces, select the **Enable on all interfaces** option on the Services tab.
- To enable HTTP on select interfaces, clear the **Enable on all interfaces** option on the Services tab, select the interface, and move it to the appropriate list by clicking the direction arrows:
 - To enable HTTP access on an interface, add the interface to the Selected interfaces list.
 - To disable HTTP access on an interface, add the interface to the Available interfaces list.

► HTTPS Web Access:

- Enable HTTPS Access: To enable HTTPS access, select the **Enable HTTPS access** option on the Services tab.
- HTTPS Certificate: Specifies SSL certificates to be used for encryption. This field is available only after you create an SSL certificate, as described in “Generating SSL certificates” on page 122. To specify the HTTPS certificate, select a certificate from the HTTPS certificate list on the Services tab.
- Enable HTTPS on All Interfaces: To enable HTTPS on all interfaces, select the **Enable HTTPS on all interfaces** option on the Services tab.

- To enable HTTPS on select interfaces, clear the **Enable on all interfaces** option on the Services tab, select the interface, and move it to the appropriate list by clicking the direction arrows:
 - To enable HTTPS access on an interface, add the interface to the Selected interfaces list.
 - To disable HTTPS access on an interface, add the interface to the Available interfaces list.

JUNOScript over SSL: JUNOScript over SSL can be used to enable secured SSL access to the JUNOScript XML scripting API. A JUNOScript client such as JUNOScope is required.

After selecting the appropriate options, click **OK** to apply the configuration.

To verify that web access is enabled correctly, connect to the router using one of the following methods:

- ▶ For HTTP access: In your Web browser, type `http://URL` or `http://IP address`.
- ▶ For HTTPS access: In your Web browser, type `https://URL` or `https://IP address`.

6.1.2 Dashboard tab

When you first get into the J-Web, you will see a chassis view of the device, the Dashboard tab, which is the default tab of J-Web, as shown in Figure 6-3.

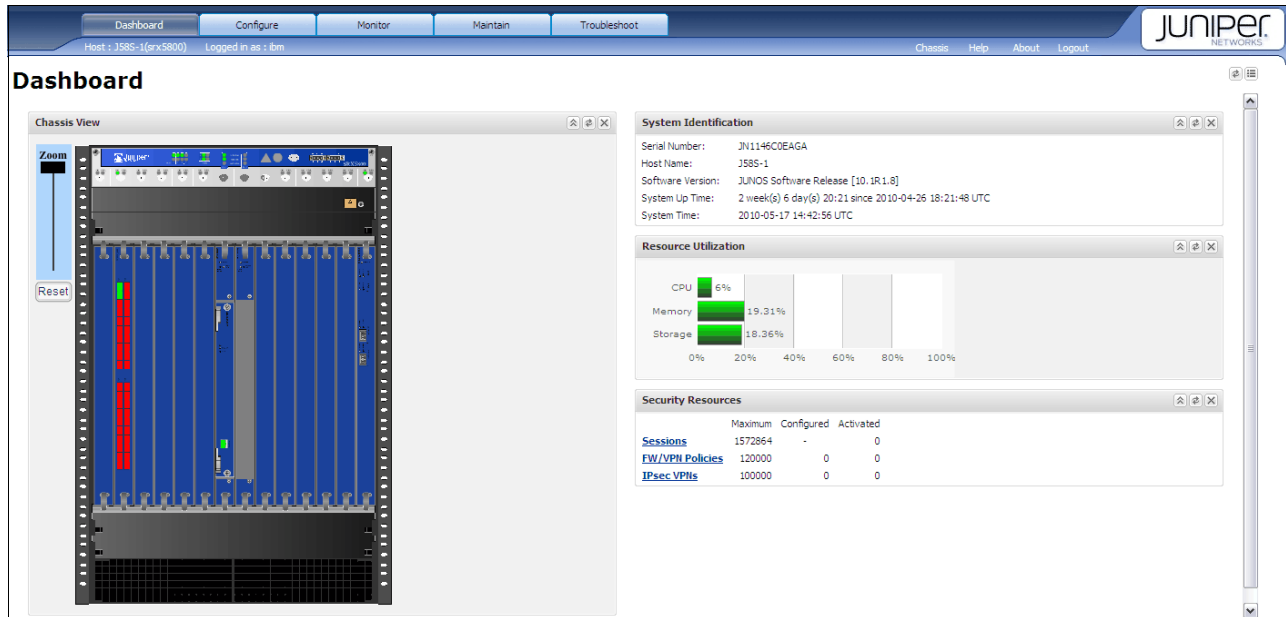


Figure 6-3 Dashboard Tab

In the Dashboard tab, you can view:

- ▶ Graphical Chassis: Shows the port status of each slot on the front view of the chassis. You can slide the zoom adjuster to the desired view. When you hover over the installed slot, a pop-up box displays the serial number, version, and description of that slot.
- ▶ System Identification box: Shows the information, such as serial number, host name, name, software version, system up time, and system time related to the system.
- ▶ Resource Utilization box: Shows the CPU, memory, and storage usage of the system.
- ▶ Security Resources box: Shows the maximum number of sessions, FW/VPN policies, and IPsec VPNs that can be configured on the system as well as the number of configured and activated settings.

6.1.3 Configure tab

The Configure tab, Figure 6-4, gives you an easy-to-use interface to configure and view your Ethernet Appliance.

Host : J58S-1(srx5800) Logged in as : ibm

Configure

Interfaces

Interface Name	Link State	Configured	Description
ge-1/0/0	Up	No	Gigabit Ethernet Interface 'ge-1/0/0'
ge-1/0/1	Up	No	Gigabit Ethernet Interface 'ge-1/0/1'
ge-1/0/2	Down	No	Gigabit Ethernet Interface 'ge-1/0/2'
ge-1/0/3	Down	No	Gigabit Ethernet Interface 'ge-1/0/3'
ge-1/0/4	Down	No	Gigabit Ethernet Interface 'ge-1/0/4'
ge-1/0/5	Down	No	Gigabit Ethernet Interface 'ge-1/0/5'
ge-1/0/6	Down	No	Gigabit Ethernet Interface 'ge-1/0/6'
ge-1/0/7	Down	No	Gigabit Ethernet Interface 'ge-1/0/7'
ge-1/0/8	Down	No	Gigabit Ethernet Interface 'ge-1/0/8'
ge-1/0/9	Down	No	Gigabit Ethernet Interface 'ge-1/0/9'
ge-1/1/0	Down	No	Gigabit Ethernet Interface 'ge-1/1/0'
ge-1/1/1	Down	No	Gigabit Ethernet Interface 'ge-1/1/1'
ge-1/1/2	Down	No	Gigabit Ethernet Interface 'ge-1/1/2'
ge-1/1/3	Down	No	Gigabit Ethernet Interface 'ge-1/1/3'
ge-1/1/4	Down	No	Gigabit Ethernet Interface 'ge-1/1/4'
ge-1/1/5	Down	No	Gigabit Ethernet Interface 'ge-1/1/5'
ge-1/1/6	Down	No	Gigabit Ethernet Interface 'ge-1/1/6'
ge-1/1/7	Down	No	Gigabit Ethernet Interface 'ge-1/1/7'
ge-1/1/8	Down	No	Gigabit Ethernet Interface 'ge-1/1/8'
ge-1/1/9	Down	No	Gigabit Ethernet Interface 'ge-1/1/9'
ge-1/2/0	Down	No	Gigabit Ethernet Interface 'ge-1/2/0'
ge-1/2/1	Down	No	Gigabit Ethernet Interface 'ge-1/2/1'
ge-1/2/2	Down	No	Gigabit Ethernet Interface 'ge-1/2/2'
ge-1/2/3	Down	No	Gigabit Ethernet Interface 'ge-1/2/3'

Figure 6-4 Configure tab

On the left side of Figure 6-4 is the configuration hierarchy. When you click one of the configuration hierarchies, the information about the hierarchy is displayed on the main portion of the window. Here you can view or edit the configuration quickly and easily without configuring each statement individually.

CLI Tools, on the bottom of the configuration hierarchy, gives you a flexible text editor that functions similarly to the CLI of Junos.

6.1.4 Monitor tab

On the Monitor tab, Figure 6-5, you can view real-time information and statistics.

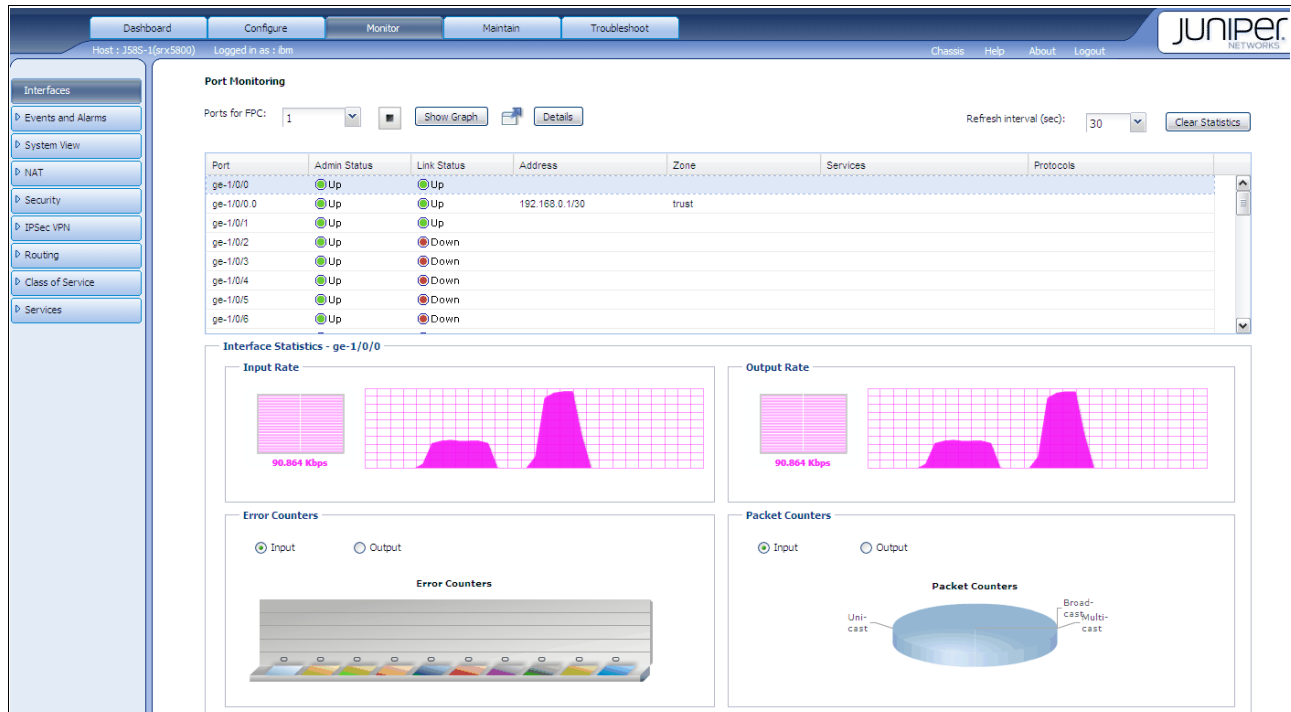


Figure 6-5 Monitor tab

For example, the interface hierarchy provides statistics in graphical and colorful pie charts and graphs. If you move the mouse pointer to various parts of the window, you are presented with more detailed information.

6.1.5 Maintain tab

The Maintain tab, Figure 6-6, provides easy file and software management functions.

Host : J58S-1(srx5800) Logged in as : ibm

Files

Clean Up Files

If you are running low on storage space on your device, you can click on the "Clean Up Files" button below. By doing so, the device will perform the following:

- Rotate your log files
- Delete log files in /var/log that are not currently being written to
- Delete temporary files in /var/tmp that have not been touched in 2 days
- Delete all crash files in /var/crash
- Delete all old software *.tgz files in /var/sw/pkg

Alternatively, you can click on the "File Type" group name below to manually download and delete individual files.

[Clean Up Files](#)

Download and Delete Files

File Type	Directory	Usage
Log Files	/var/log	1.2M
Temporary Files	/var/tmp	176M
Jailed Temporary Files (Install, Session, etc)	/var/jail/tmp	28K
Old JUNOS Software	/var/sw/pkg	180M
Crash (Core) Files	/var/crash	2.0K
Database files	/var/db	3.8M

Delete Backup JUNOS Package

There is no backup JUNOS package.

Figure 6-6 Maintain Tab

You can perform maintenance of your Ethernet devices with the following options:

- ▶ **Files:** You can download and delete log files and temporary files to keep the compact-flash device from becoming full.
- ▶ **Configuration Management:** Using this option, you can retrieve historical configuration files and to compare the differences between configurations.
- ▶ **Software:** Gives you easy ways to upgrade and downgrade Junos software.
- ▶ **Licenses:** Shows you the installed licenses and feature summary.
- ▶ **Reboot:** Use this option to schedule various methods of rebooting the switch.
- ▶ **Snapshot™:** Use this option to configure storage devices to replace the primary boot device on your router or to act as a backup boot device.

6.1.6 Troubleshoot tab

The Troubleshoot tab, Figure 6-7, provides tools for basic troubleshooting without the need to open a CLI session.

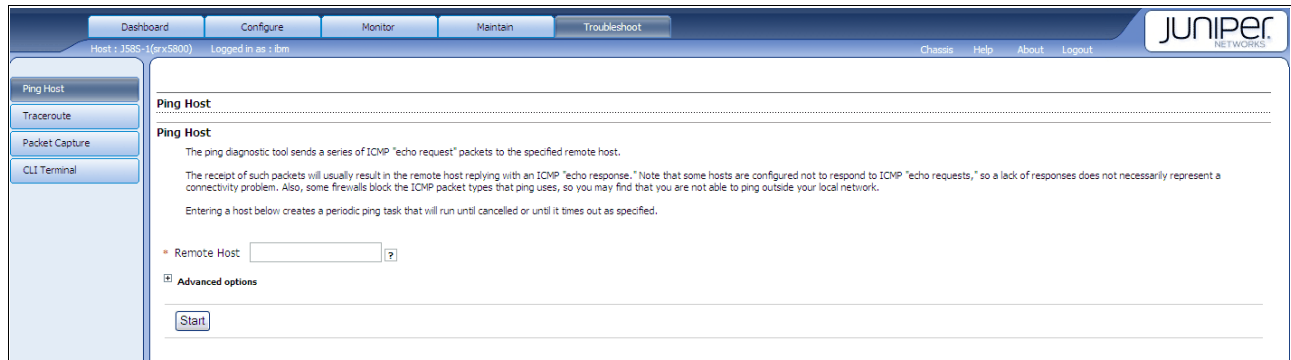


Figure 6-7 Troubleshoot tab

You can use quick and useful troubleshooting tools, such as Ping, Traceroute, Packet Capture, and CLI Terminal to investigate the network problem.

6.2 Junos CLI

Junos CLI is a Juniper Networks-specific command shell that runs on top of a UNIX-based operating system kernel. The CLI provides command help and command completion.

The CLI also provides a variety of UNIX utilities, such as Emacs-style keyboard sequences, that allow you to move around on a command line and scroll through recently executed commands, regular expression matching to locate and replace values and identifiers in a configuration, filter command output, or log file entries, store and archive router files on a UNIX-based file system, and exit from the CLI environment and create a UNIX C shell or Bourne shell to navigate the file system, manage switch processes, and so on.

6.2.1 CLI modes

The CLI has two modes:

- ▶ Operational mode
- ▶ Configuration mode

In the Operational mode, you can troubleshoot and monitor the software, router, and network. The right angle bracket (>) is the identifier in the operational mode, as shown in the following example:

```
user@J48E>
```

Configuration mode, also known as Edit mode, is where you edit configurations and where the actual statements for interfaces, routing protocols, and others are placed. The pound character (#) identifies configuration mode, as shown in the following example:

```
user@J48E#
```

6.2.2 Logging in CLI

Because Junos runs on top of a UNIX environment, you require a username and password to log in. A root user is created by default. You log into CLI using root for first time and then create other user accounts and assigns permissions.

When you first enter the routers and switches through Telnet, SSH, or direct console access, a login prompt is displayed. After entering the correct username and password, you are placed directly into the operational mode, as shown in Example 6-1.

Example 6-1 Logging in CLI

```
J48E (tty0)
```

```
login: ibm
```

```
Password:
```

```
--- JUNOS 10.1R1.8 built 2010-02-12 17:24:20 UTC
```

```
{master:0}
```

```
ibm@J48E>
```

An exception case of being placed into operational mode is when you log in as root user. In this case, you are actually placed in the shell (designated by the percent (%) sign) and must start the CLI manually using the `cli` command shown in Example 6-2.

Example 6-2 Logging in CLI using root user

```
J48E (tty0)

login as: root
root@10.1.1.10's password:

--- JUNOS 10.1R1.8 built 2010-02-12 17:24:20 UTC

root@J48E:RE:0% cli
{master:0}
root@J48E>
```

6.2.3 CLI operational mode

Operational mode CLI commands monitor and control the operation of the routers and switches. The operational mode CLI commands are hierarchically structured, as shown in Figure 6-8.

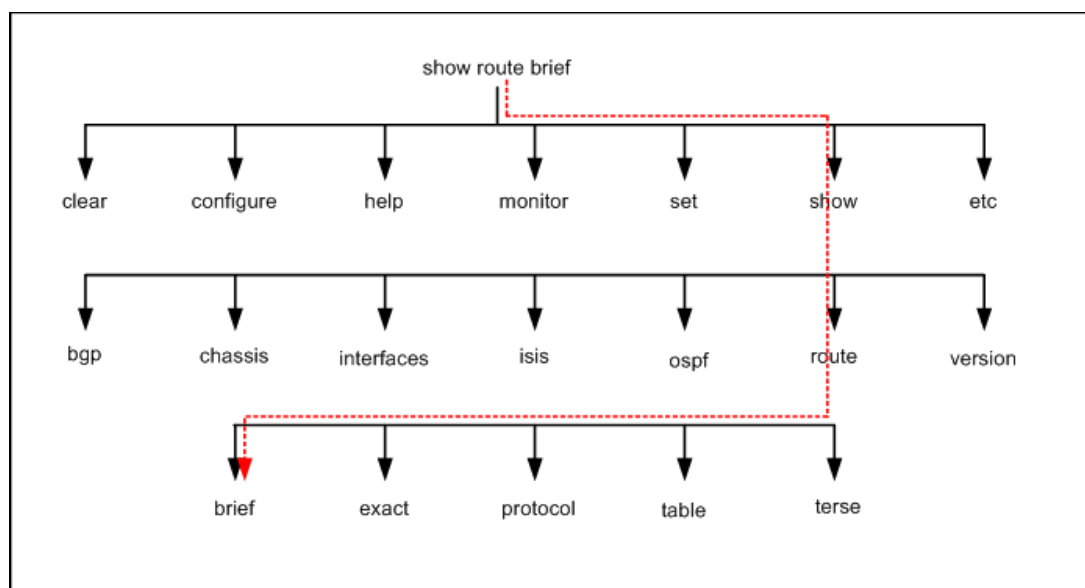


Figure 6-8 Hierarchy of commands

The commands are mainly executed from the default CLI level (`user@switch>`). For example, to see the brief routes of the current routing table, type `show route brief`, as shown in Example 6-3.

Example 6-3 Output of show route brief

```
ibm@J48E> show route brief

inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```

10.1.1.0/24      *[Direct/0] 1d 00:17:14
                  > via me0.0
10.1.1.10/32     *[Local/0] 1d 22:53:26
                  Local via me0.0
192.168.1.0/30   *[Direct/0] 1d 03:50:53
                  > via ge-0/0/0.0
192.168.1.1/32   *[Local/0] 1d 03:50:53
                  Local via ge-0/0/0.0
224.0.0.2/32     *[PIM/0] 1d 22:53:28
                  MultiRecv
224.0.0.13/32    *[PIM/0] 1d 22:53:28
                  MultiRecv

inet6.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

ff02::2/128      *[PIM/0] 1d 22:53:28
                  MultiRecv
ff02::d/128      *[PIM/0] 1d 22:53:28
                  MultiRecv

```

Key operational-mode capabilities

Junos software offers useful key operational-mode capabilities that include the following:

- ▶ Entering configuration mode
- ▶ Controlling the CLI environment
- ▶ Existing CLI
- ▶ Monitoring and troubleshooting
 - clear
 - monitor
 - ping
 - show
 - test
 - traceroute
- ▶ Connecting to other network systems
- ▶ Copy files
- ▶ Restarting software processes
- ▶ Performing system-level operations

Command completion

The command completion feature saves you lots of time and energy because it provides syntax checking as you type. Gone are the days when a command is typed on a line. After pressing Enter, the command is either invalid or not supported on that version of software. In Junos, any error or ambiguity is detected early, and the router and switch presents a list of possible valid completion.

Hidden commands and variable names (for example, group names) cannot be completed using the Spacebar. The entire command must be manually typed, for example:

```
user@J48E> show groups junos-defaults
```


Command completion is accomplished using either the Spacebar or the Tab key. To complete a command, statement, or option that you partially typed, press the Tab key or the Spacebar. If the partially typed letters uniquely identify a command, the complete command name appears. Otherwise, a beep indicates that you entered an ambiguous command and the possible completions are displayed. This completion feature also applies to other strings, such as filenames, interface names, usernames, and configuration statements.

For example, to view the configuration of a certain Gigabit Ethernet interface, you can type the command as shown in Example 6-4.

Example 6-4 Using Spacebar and tab key

```
user@J48E> sh<space>ow conf<space>iguration int<space>erfaces g<tab>e-0/0/0
<enter>
```

Notice that the Spacebar is used until a variable is reached, and the interface name is used when the Tab key must be used (the Spacebar completes only commands and not variables).

In Example 6-4, the syntax checker goes word-by-word each time the Spacebar or Tab key is used, and minimum characters are typed to avoid ambiguity. In the case where the syntax checker notices that an error is an incomplete word, it states the ambiguity and provides the possible completions, as shown in Example 6-5.

Example 6-5 Using Spacebar to list possible completions

```
ibm@J48E> show ip<space>
      ^
'ip' is ambiguous.
Possible completions:
  ip-source-guard      Show IP source guard information
  ipsec                Show IP Security information
  ipv6                 Show IP version 6 information
```

Emacs

Junos software defaults to a VT100 terminal type. This terminal type enables the user of keyboard Arrow keys with any additional configuration modification. With this terminal type, you can use Emacs-style keystrokes that allow you to move the cursor around the command line, edit the command line or delete specific characters or words. Some useful Emacs keystrokes are:

Ctrl+B	Moves the cursor left one character.
Ctrl+A	Moves the cursor to the beginning of the command line.
Ctrl+F	Moves the cursor right one character.
Ctrl+E	Moves the cursor to the end of the command line.
Ctrl+D	Deletes the character over the cursor.
Ctrl+K	Deletes from the cursor to the end of the line.
Ctrl+U	Deletes all the characters and negates the current command.
Ctrl+W	Deletes the entire word to the left of the cursor.
Ctrl+L	Redraws the current line.

Ctrl+P	Scrolls backward through the previously typed commands. You also can use the Up arrow for this purpose.
Ctrl+N	Scrolls forward through the previously typed commands. You also can use the Down arrow for this purpose.
Ctrl+R	Search the previous CLI history for a search string.

Pipe commands

Another useful feature of the Junos CLI is the use of pipe commands to control the output of any command. When a command, such as **show** is issued, the data is placed into a buffer and is displayed when the Enter key is pressed. A pipe command can be used to alter the display buffer.

The following sections examine the most common application of pipe commands:

The count command

Use the **count** command to count the lines in the output, as shown in Example 6-6.

Example 6-6 Pipe with count command

```
ibm@J48E> show configuration | count
Count: 352 lines
```

The display command

Use the **display** command to show additional kinds of information, such as set commands, as shown in Example 6-7.

Example 6-7 Pipe with display set command

```
ibm@J48E> show configuration | display set
set version 10.1R1.8
set system host-name J48E
set system root-authentication encrypted-password
"$1$U7AW/vKF$UUNnJTiq3psmzUh1mRte/0"
set system login user ibm uid 2001
set system login user ibm class super-user
set system login user ibm authentication encrypted-password
"$1$qb66ubMo$bhUmZ/iY3aPYEPCUuaWEn."
set system services ssh
set system services telnet
set system services web-management http
set system syslog user * any emergency
set system syslog file messages any notice
set system syslog file messages authorization info
set system syslog file interactive-commands interactive-commands any
set interfaces ge-0/0/0 unit 0 family inet address 192.168.1.1/30
set interfaces ge-0/0/1 unit 0 family ethernet-switching
set interfaces ge-0/0/2 unit 0 family ethernet-switching
set interfaces ge-0/0/3 unit 0 family ethernet-switching
set interfaces ge-0/0/4 unit 0 family ethernet-switching
```

The display detail command

The **display detail** command shows configuration data in detail, and it is useful to check if there is any limitation on any configuration element that is defined. Example 6-8 shows the group name range for characters between the range 1... 254.

Example 6-8 Pipe with display detail

```
ibm@J48E>> show configuration | display detail
## Last commit: 2010-06-27 17:20:58 PDT by root
##
## version: Software version information
## require: system
##
version "10.2I0 [builder]";
##
## groups: Configuration groups
## require: configuration
##
groups {
  ##
  ## Group name
  ## range: 1 .. 254
  ##
  node0 {
    ##
    ## system: System parameters
    ## require: admin system
    ##
    system {
      ##
      ## host-name: Hostname for this router
      ## range: 0 .. 255
      ## match (regex): ^[[:alnum:]]+_$
      ## require: system
      ##
      host-name SRX3400-1;
      ##
      ## backup-router: IPv4 router to use while booting
      ## alias: inet-backup-router
      ## require: system
      ## Address of router to use while booting
      ##
      ##
      ## destination: Destination network reachable through the router
      ##
      backup-router 172.19.100.1 destination 0.0.0.0/0;
      ##
      ## time-zone: Time zone name or POSIX-compliant time zone string
      ## units: <continent>/<major-city> or <time-zone>
      ##
      ## default: UTC
      ##
      ## saved-core-files: Number of saved core files per executable
      ## range: 1 .. 10
      ##
    }
  }
}
```

The except command

The **except** command shows only text that does not match a pattern, as shown in Example 6-9.

Example 6-9 Pipe with except command

```
ibm@J48E> show interfaces terse | except ge
Interface           Admin Link Proto  Local              Remote
vcp-0               up    down
vcp-0.32768         up    down
vcp-1               up    down
vcp-1.32768         up    down
bme0                up    up
bme0.32768          up    up   inet    128.0.0.1/2
                                   128.0.0.16/2
                                   128.0.0.32/2
                                   tnp    0x10
dsc                 up    up
gre                 up    up
ipip                up    up
lo0                 up    up
lsi                 up    up
me0                 up    up
me0.0               up    up   inet    10.1.1.10/24
mtun                up    up
pimd                up    up
pime                up    up
tap                 up    up
vlan                up    up
vlan.0              up    up   inet
vme                 up    down
vme.0               up    down inet
```

The find command

The **find** command searches for the first occurrence of a pattern, as shown in Example 6-10

Example 6-10 Pipe with find command

```
ibm@J48E> show interfaces ge-0/0/0 extensive | find traffic
Traffic statistics:
Input bytes :          427368738          0 bps
Output bytes :          430971601          0 bps
Input packets:          542474           0 pps
Output packets:          550223           0 pps
IPv6 transit statistics:
Input bytes :              0
Output bytes :              0
Input packets:              0
Output packets:              0
Input errors:
Errors: 0, Drops: 0, Framing errors: 0, Runt: 0, Policed discards: 0, L3
incompletes: 0, L2 channel errors: 0,
L2 mismatch timeouts: 0, FIFO errors: 0, Resource errors: 0
Output errors:
```

```
Carrier transitions: 13, Errors: 0, Drops: 0, Collisions: 0, Aged packets: 0,
FIFO errors: 0,
HS link CRC errors: 0, MTU errors: 0, Resource errors: 0
Egress queues: 8 supported, 4 in use
Queue counters:      Queued packets  Transmitted packets      Dropped packets
0 best-effort        0                543076                    0
1 assured-forw       0                0                        0
5 expedited-fo       0                0                        0
7 network-cont       0                7147                     0
Active alarms   : None
Active defects  : None
```

The match command

The **match** command shows only text that matches a pattern, as shown in Example 6-11.

Example 6-11 Pipe with match command

```
ibm@J48E> show log messages | match "feb 13"
Feb 13 00:08:45 J48E chassisd[802]: CHASSISD_SNMP_TRAP6: SNMP trap generated:
Power Supply failed (jnxContentsContainerIndex 2, jnxContentsL1Index 1,
jnxContentsL2Index 2, jnxContentsL3Index 0, jnxContentsDescr Power Supply 1,
jnxOperatingState/Temp 6)
Feb 13 01:08:45 J48E chassisd[802]: CHASSISD_SNMP_TRAP6: SNMP trap generated:
Power Supply failed (jnxContentsContainerIndex 2, jnxContentsL1Index 1,
jnxContentsL2Index 2, jnxContentsL3Index 0, jnxContentsDescr Power Supply 1,
jnxOperatingState/Temp 6)
Feb 13 02:08:46 J48E chassisd[802]: CHASSISD_SNMP_TRAP6: SNMP trap generated:
Power Supply failed (jnxContentsContainerIndex 2, jnxContentsL1Index 1,
jnxContentsL2Index 2, jnxContentsL3Index 0, jnxContentsDescr Power Supply 1,
jnxOperatingState/Temp 6)
```

Context-sensitive help

The Junos CLI provides context-sensitive help at any point in a command line. Help tells you which options are acceptable at the current point in the command and provides a brief description of each command or command option.

To receive help at any time while in the Junos CLI, type a question mark (?). You do not need to press Enter. See Example 6-12.

Example 6-12 Help with question mark

```
ibm@J48E> ?
Possible completions:
clear                Clear information in the system
configure            Manipulate software configuration information
file                 Perform file operations
help                 Provide help information
load                 Load information from file
.....

ibm@J48E> clear ?
Possible completions:
arp                  Clear address resolution information
```

auto-configuration	Clear auto-configuration action
bfd	Clear Bidirectional Forwarding Detection information
bgp	Clear Border Gateway Protocol information
captive-portal	Clear 802.1X session
chassis	Clear chassis information

Help topic

There are several ways to use the **help** command. The **help topic** command displays usage guidelines for the statement. See Example 6-13.

Example 6-13 Help topic

```
ibm@J48E> help topic interfaces ?
Possible completions:
  accept-data          Accept packets destined for virtual address
  accept-source-mac    Policers for specific source MAC addresses
  access-profile-chap  CHAP profile associated with physical interface
  accounting           Packet counting for transit traffic
  accounting-profile   Accounting profile
  acfc                Compression of Address and Control fields in PPP header
  acknowledge-retries  Setting for link acknowledgment messages
  acknowledge-timer    Setting for link acknowledgment messages
  action-red-differential-delay Setting for link services differential delay
  address              Interface address and destination prefix

.....

ibm@J48E> help topic interfaces address
                        Configuring the Interface Address

You assign an address to an interface by specifying the address when
configuring the protocol family. For the inet family, configure the
interface's IP address. For the iso family, configure one or more
addresses for the loopback interface. For the ccc, tcc, mpls, tnp, and
vpls families, you never configure an address.

.....
```

Help reference

The **help reference** command displays summary information for the statement, as shown in Example 6-14.

Example 6-14 Help reference

```
ibm@J48E> help reference interfaces address
                        address

Syntax

address address {
    arp ip-address (mac | multicast-mac) mac-address <publish>;
    broadcast address;
```

```
destination address;
.....
Hierarchy Level

[edit interfaces interface-name unit logical-unit-number family family],

[edit logical-systems logical-system-name interfaces interface-name unit
logical-unit-number family family]

Release Information

Statement introduced before JUNOS Release 7.4.

Description

Configure the interface address.
```

6.2.4 CLI configuration mode

You configure Junos software by entering configuration mode. Within configuration mode, you can configure all features of Junos software, such as interfaces, routing protocols, user access, routing policy as well as chassis properties.

Active and candidate configuration

Junos software has two configuration files that are *candidate configuration* and *active configuration*. The active configuration is the current running configuration in Junos software; whereas, the candidate configuration is the temporary text file that is being modified while in configuration mode.

The candidate configuration becomes the active configuration when the **commit** command is issued, and there are no syntax errors detected. The old, active configuration is saved as a file called rollback 1.

In case there is a mistake during the changes, you can easily recover the old active configuration by issuing the **rollback 1** command. The old active configuration replaces the candidate configuration. You must issue a **commit** command to activate the rollback file.

Junos software saves not only the last active configuration but also the previous 49 configurations. Every time a **commit** command is issued, the saved file moves down to the list of 49. The first **commit** creates rollback1. The second commit becomes rollback 1, and the old rollback then becomes rollback 2. The process of rollback is illustrated in Figure 6-9 on page 140.

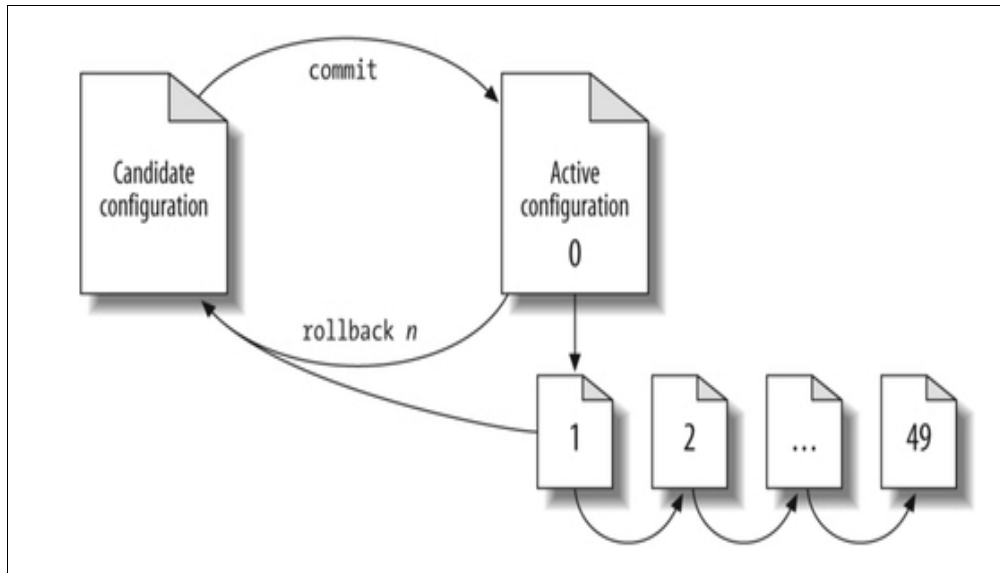


Figure 6-9 Configuration rollback process

Entering configuration mode

To actually configure the router, enter configuration mode by typing the word `configure` or `edit` in operation mode. The router prompt changes to the pound (#) character, as shown in Example 6-15.

Example 6-15 Entering configuration mode

```
ibm@J48E> configure
Entering configuration mode
```

```
{master:0} [edit]
ibm@J48E#
```

.....

```
ibm@J48E> edit
Entering configuration mode
```

```
{master:0} [edit]
ibm@J48E#
```

- ▶ *Configure exclusive:* Allows only a single user to configure the router. In exclusive mode, no other users can make changes to the configuration besides the single user that entered exclusively.
- ▶ *Configure private:* Allows multiple users to configure different pieces of the configuration. Using private mode, each user gets a copy of the current configuration and only changes that they make are applied. If two users attempt to make the same change, such as changing an IP address to the same interface, the change is rejected and both users exit configuration mode to resolve their conflict.

Junos hierarchical configuration

The CLI is actually composed of many directories and subdirectories that builds a hierarchical configuration. So, when you issue the `set system services telnet` command, the system

directory is accessed, followed by the subdirectory `services`, and ending in the command `telnet` to enable the Telnet service, as shown in Figure 6-10.

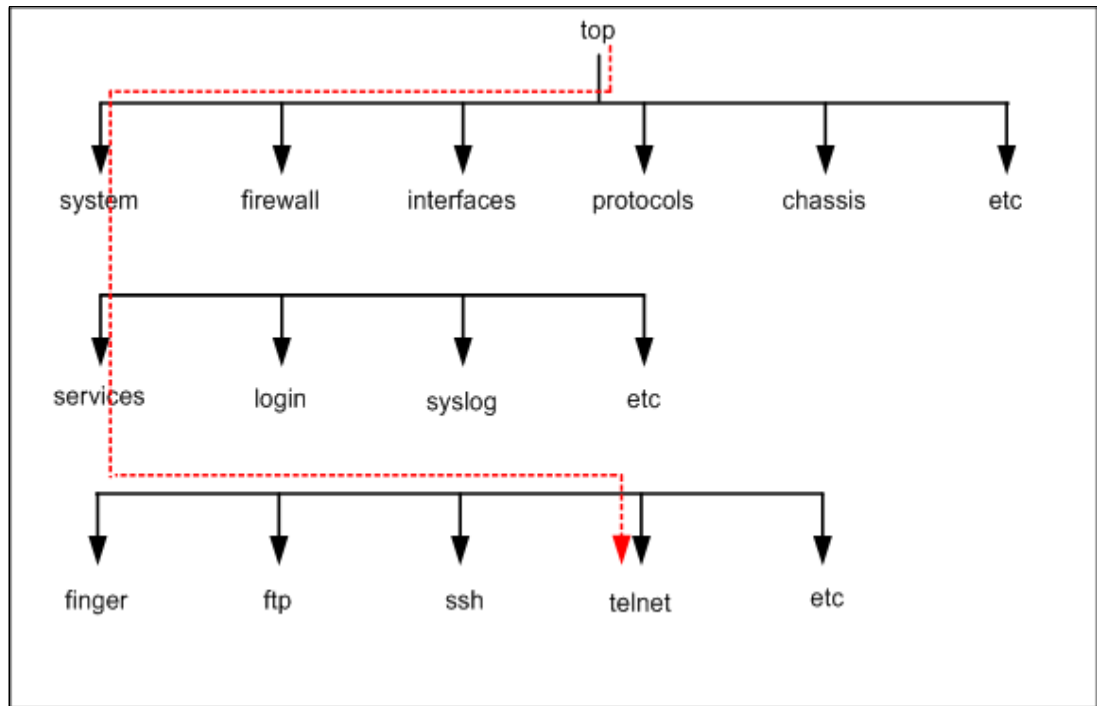


Figure 6-10 Junos configuration hierarchy

Moving between levels

You can move between levels using the `edit` command, which functions like a changing directory (CD) command, for example, move from the edit directory to the protocol subdirectory, and so on in Figure 6-11 on page 142.

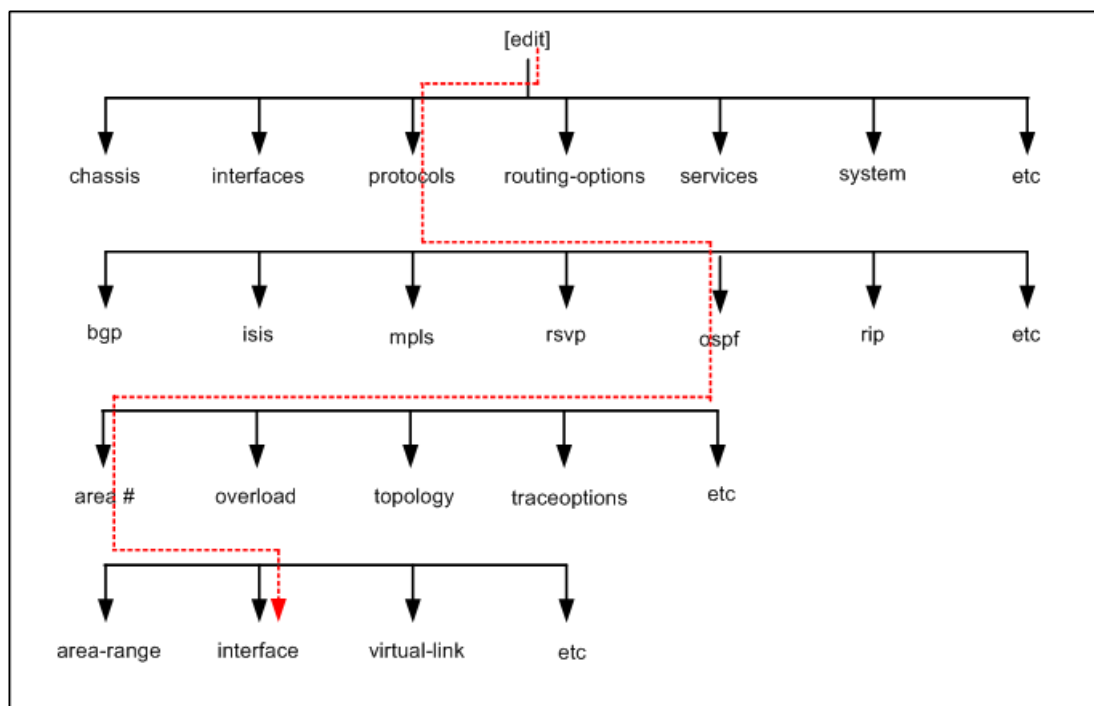


Figure 6-11 Changing directory with edit

Example 6-16 shows how to use **edit** to move to the lower level of ospf.

Example 6-16 Moving to another directory

```
{master:0} [edit]
ibm@J48E# edit protocols ospf area 0 interface ge-0/0/0

{master:0} [edit protocols ospf area 0.0.0.0 interface ge-0/0/0.0]
ibm@J48E#
```

The **up** command moves up one level in the hierarchy, for example, to move up one level as in Figure 6-12 on page 143.

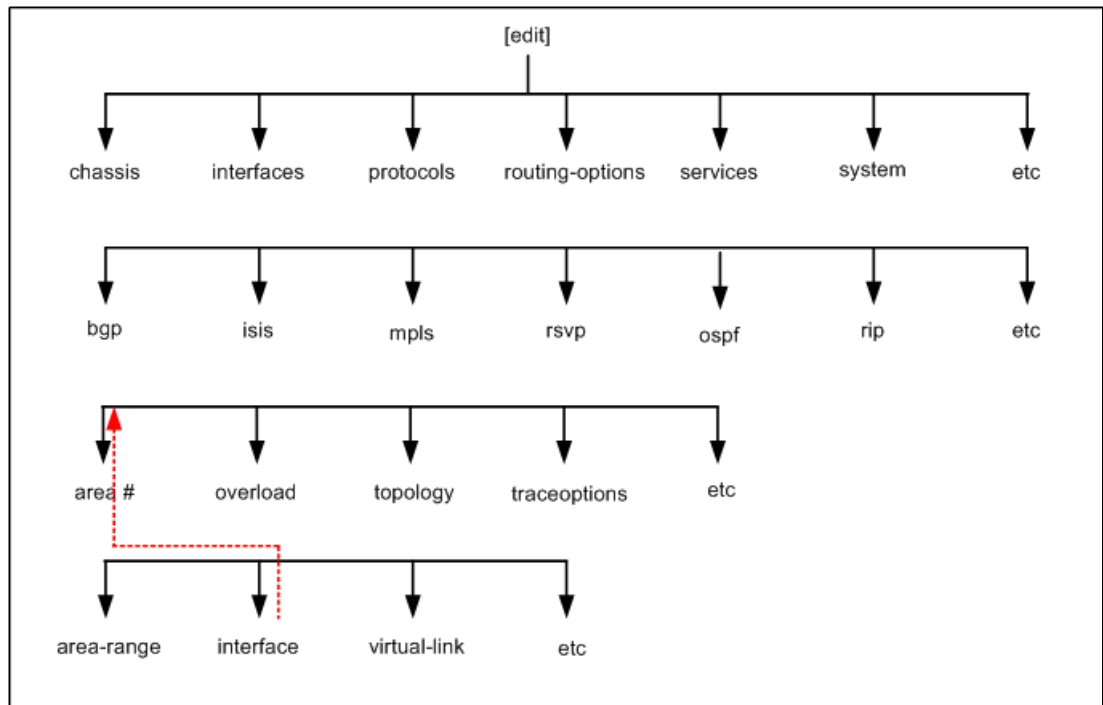


Figure 6-12 Move up one level

Example 6-17 shows moving up one level.

Example 6-17 Moving up one level

```

{master:0}[edit protocols ospf area 0.0.0.0 interface ge-0/0/0.0]
ibm@J48E# up

{master:0}[edit protocols ospf area 0.0.0.0]
ibm@J48E#
  
```

The **up n** command moves up n levels, for example, move up two levels, as shown in Figure 6-13 on page 144.

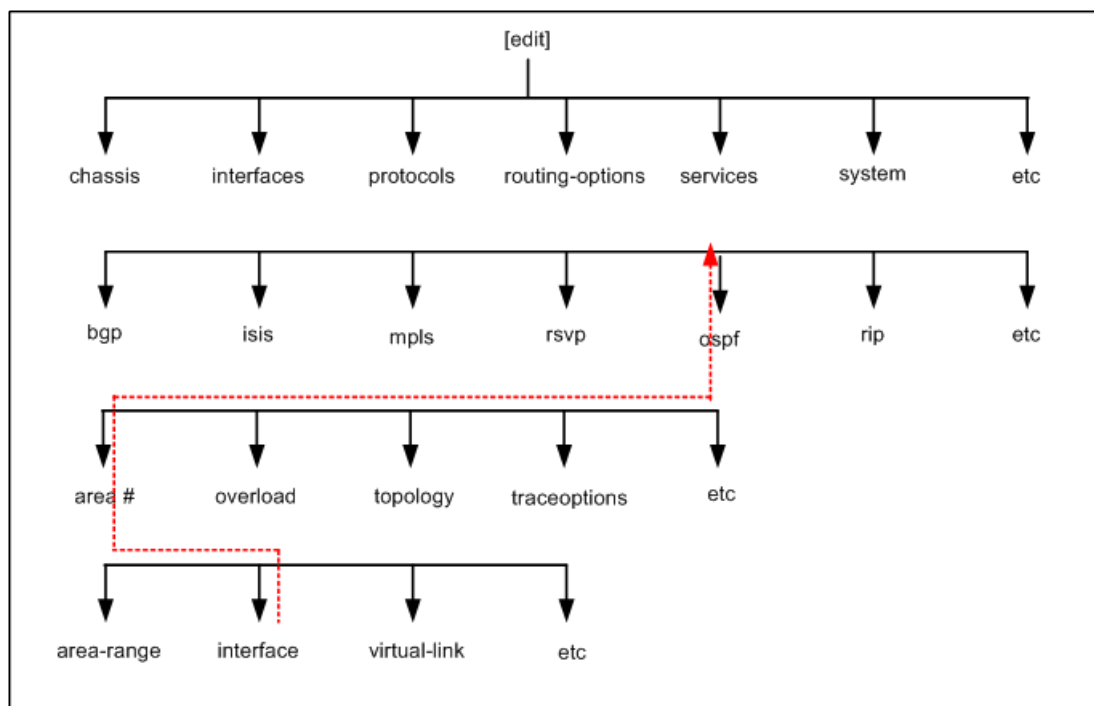


Figure 6-13 Moving up two level

Example 6-18 shows moving up two level.

Example 6-18 Moving up two level

```
{master:0}[edit protocols ospf area 0.0.0.0 interface ge-0/0/0.0]
ibm@J48E# up 2
```

```
{master:0}[edit protocols ospf]
ibm@J48E#
```

The **top** command moves to the top of the hierarchy, for example, moves to the top of the hierarchy, as shown in Figure 6-14 on page 145.

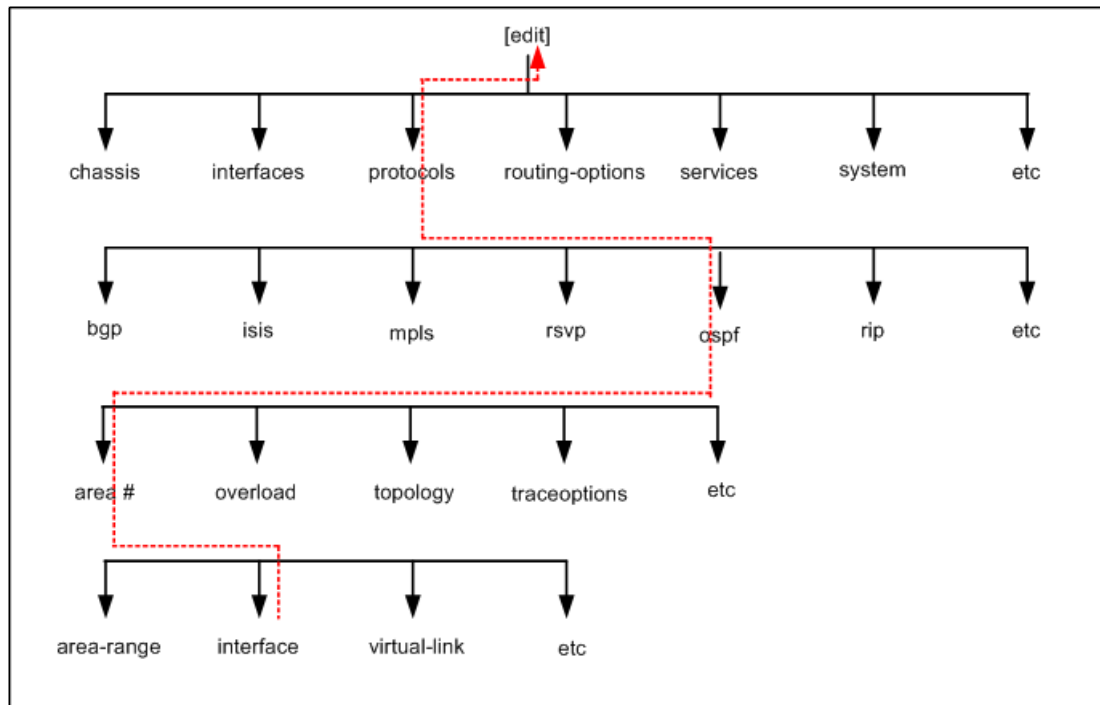


Figure 6-14 Moving to the top of the hierarchy

Example 6-19 shows moving to the top the hierarchy.

Example 6-19 Moving to the top of the hierarchy

```
{master:0}[edit protocols ospf area 0.0.0.0 interface ge-0/0/0.0]
ibm@J48E# top

{master:0}[edit]
ibm@J48E#
```

Delete command

The opposite of the **set** command to remove configuration from the router is the **delete** command. Usually this command removes a single line. You can also use it to remove an entire hierarchy.

For example, to remove the Telnet service from the router, change the **previous set** command to a **delete** command, as shown in Example 6-20.

Example 6-20 Delete configuration

```
ibm@J48E# delete system services telnet
```

Viewing candidate configuration

To display the candidate configuration, use the **show** command. This command displays the configuration at the current hierarchy level or at the specified level below the current location.

For example, you can display only portions that you want to view from the root hierarchy. See Example 6-21 on page 146.

Example 6-21 Show candidate configuration

```
ibm@J48E# show system services
ftp;
ssh;
telnet;
```

```
[edit]
ibm@J48E#
```

The commit command

The router and switch do not automatically apply the configuration. You must use the **commit** command to activate your candidate configuration.

There are several options of the **commit** command:

commit	Just commit current configuration
commit check	Check correctness of syntax but do not apply changes
commit confirmed	Temporarily active the configuration
commit at	Schedule to activate the changes at the specified time
commit comment	Add comments to the commit
commit and-quit	Activate the changes and exit configuration mode in a single step

The rollback command

The Junos software saves the last 50 committed versions of the configuration. To overwrite or restore the candidate configuration with one of these previously committed configurations, use the **rollback** command:

- ▶ When you use only the **rollback** or **rollback 0** command, it resets the candidate configuration to the currently active configuration.
- ▶ When you use the **rollback 1** command, it loads the previously active configuration.
- ▶ When you use the **rollback n** command (n can be a number in the range 0 through 49), it loads the referenced rollback version of configuration.
- ▶ The **rollback** command modifies only the candidate configuration. You must issue the **commit** command to activate the changes.

Saving a configuration

You can save the candidate configuration to an ASCII file. By default the files are stored under `/var/home` directory under current USER directory name. You can provide full path name if you prefer to save the file in other directory.

The **Save** command only saves the configuration in its current hierarchy and below. To save the entire candidate configuration, you must start at the top level of the configuration hierarchy. We recommend that you save a file using a meaningful filename, as shown in Example 6-22.

Example 6-22 Save configuration

```
ibm@J48E# save config-backup-2010-04-29
Wrote 261 lines of configuration to 'config-backup-2010-04-29'
```

```
[edit]
ibm@J48E#
```

Loading a configuration

You can retrieve the previously saved configuration by using the **load** command, which has several options for specifics of the operation:

load override filename Completely overrides an existing configuration

load merge filename Combines the current configuration with the configuration being loaded.

load replace filename Replaces existing statements in the current configuration.

load factory-default Loads the factory-default configuration.

After the load operation is completed, you must execute a **commit** command to activate the changes of the configuration.

Comparing configurations

Junos software has a useful **pipe** command that allows you to compare two files with the **compare** command. This command allows any two files, including rollback files, active files, and candidate files, to be compared and displays the differences, for example, the candidate and active configurations can be compared, as shown in Example 6-23.

Example 6-23 Compare active and candidate configuration

```
ibm@J48E# show | compare
[edit system]
+ tacplus-server {
+   10.1.1.1;
+ }
[edit system services]
- ftp;
+ ssh;

[edit]
```

Example 6-24 shows how you can compare active configuration with rollback configurations or saved files.

Example 6-24 Compare active configuration with rollback and saved files

```
[edit]
ibm@J48E> show configuration| compare rollback number

.....

ibm@J48E> show configuration| compare filename
```

The run command

While you are in configuration mode, you can execute operational-mode commands by using the **run** command. This command really saves your time from existing to operational mode and entering configuration mode during your configuration and verification.

For example, you can run the **ping** command from the interface directory in configuration mode, as shown in Example 6-25 on page 148.

Example 6-25 Run command

```
{master:0}[edit interfaces ge-0/0/0 unit 0]
ibm@J48E# run ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1): 56 data bytes
64 bytes from 192.168.1.1: icmp_seq=0 ttl=64 time=0.151 ms
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=0.111 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=0.297 ms
^C
--- 192.168.1.1 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.111/0.186/0.297/0.080 ms

{master:0}[edit interfaces ge-0/0/0 unit 0]
ibm@J48E#
```

Helpful configuration-mode command

There are several commands that help save your time and increase accuracy and consistency during your configuration:

rename	A configuration statement
replace	A pattern of configuration statements
copy	A configuration statement to another statement
deactivate	A configuration statement
insert	A configuration statement in another location

6.3 More information

For more information, refer to the below reference material:

- CLI Reference material with link
http://www.juniper.net/techpubs/en_US/junos10.1/information-products/topic-collections/swconfig-cli/swconfig-cli-TOC.html
- J-Web Reference material with link
http://jnpr.net/techpubs/en_US/junos10.1/information-products/topic-collections/jweb-user-guide/frameset.html



Class of service

As packet networks have evolved to deliver data, voice and video, quality of service has become increasingly important to ensure that applications are afforded the preferential treatment that is required to operate properly at all times.

In general terms, quality of service is the ability of a network to provide a defined level of service for particular traffic. The problem with such a broad definition is that it is extremely open. What constitutes service quality? What are the parameters that affect service in a packet-based network? How do you define particular traffic, or, more specifically, how do you determine which traffic is given what service?

The first question can be answered by understanding that the following metrics are considered to be standard measurements of service quality:

- ▶ End-to-end packet delay
- ▶ Delay jitter
- ▶ Available capacity
- ▶ Drop probability

Class of Service: Junos implementation of quality of service is called Class of Service (CoS)

7.1 Class of Service

When a network experiences congestion and delay, some packets must be dropped. Junos' implementation of quality of service is called Class of Service (CoS) and allows you to divide traffic into classes and offer various levels of throughput and packet loss when congestion occurs. This allows packet loss to happen according to rules that you configure.

For interfaces that carry IPv4, IPv6, and MPLS traffic, you can configure Junos CoS features to provide multiple classes of service for different applications. On the router, you can configure multiple forwarding classes for transmitting packets, define which packets are placed into each output queue, schedule the transmission service level for each queue, and manage congestion using a random early detection (RED) algorithm.

The Junos CoS features provide a set of mechanisms that you can use to provide differentiated services when best-effort traffic delivery is insufficient. In designing CoS applications, you must give careful consideration to your service needs, and you must thoroughly plan and design your CoS configuration to ensure consistency across all routers in a CoS domain. You must also consider all the routers and other networking equipment in the CoS domain to ensure interoperability among all equipment.

Because IBM routers implement CoS in hardware rather than in software, you can experiment with and deploy CoS features without adversely affecting packet forwarding and routing performance.

The purpose of this chapter is to explain the steps needed to configure the following CoS features on IBM j-type s-series and to provide examples on each section. You must be familiar with the basic topics of quality of service:

- ▶ Packet classification
- ▶ Forwarding Classes
- ▶ Policing
- ▶ Rewrite rules
- ▶ Packet loss priority
- ▶ Schedulers
- ▶ Drop profiles

7.2 Junos CoS components

CoS works by examining traffic that enters at the edge of your network. The edge routers classify traffic into defined service groups to provide the special treatment of traffic across the network. For example, voice traffic can be sent across certain links, and data traffic can use other links. In addition, the data traffic streams can be serviced differently along the network path to ensure that higher-paying customers receive better service. As the traffic leaves the network at the far edge, you can reclassify the traffic.

To support CoS, you must configure each router in the network. Generally, each router examines the packets that enter it to determine their CoS settings. These settings then dictate which packets are first transmitted to the next downstream router. In addition, the routers at the edges of the network might be required to alter the CoS settings of the packets that enter the network from the customer or peer networks.

Junos CoS configuration is based on the following components:

- Packet classification

Packet classification associates incoming packets with a particular CoS servicing level. In Junos software, classifiers associate packets with a forwarding class and loss priority and assign packets to output queues based on the associated forwarding class. Junos software supports two general types of classifiers:

- a. Behavior aggregate
- b. Multifield traffic classifiers

- Forwarding classes

Forwarding classes group the packets for transmission. Based on forwarding classes, you assign packets to output queues. Forwarding classes affect the forwarding, scheduling, and marking policies applied to packets as they transmit a switching platform. By default, four categories of forwarding classes are defined: best effort, assured forwarding, expedited forwarding, and network control.

- Policers

Policers limit traffic of a certain class to a specified bandwidth and burst size. Packets exceeding the policer limits can be discarded. You define policers with filters that can be associated with input interfaces.

- Rewrite rules

A rewrite rule sets the appropriate CoS bits in the outgoing packet, thus allowing the next downstream device to classify the packet into the appropriate service group. Rewriting, or marking, outbound packets is useful when the switch is at the border of a network and must alter the CoS values to meet the policies of the targeted peer.

- Scheduler

Each interface has multiple queues assigned to store packets. The switch determines which queue to service based on a particular method of scheduling. This process often involves determining which type of packet must be transmitted before another. You can define the priority, bandwidth, delay buffer size, and drop profiles to be applied to a particular queue for packet transmission.

A scheduler map associates a specified forwarding class with a scheduler configuration.

- Drop profiles and loss priority

Drop profile is a mechanism that defines parameters that allow packets to be dropped from the network. Drop profiles define the meanings of the loss priorities. When you configure drop profiles, you are essentially setting the value for queue fullness. The queue fullness represents a percentage of the queue that is used to store packets in relation to the total amount that has been allocated for that specific queue.

Loss priorities set the priority of dropping a packet. Loss priority affects the scheduling of a packet without affecting the packet's relative ordering. You can use the loss priority setting to identify packets that have experienced congestion. Typically you mark packets exceeding some service level with a high loss priority.

7.3 Packet flow

On IBM j-type s-series, you configure CoS functions using various components. These components are configured individually or in a combination to define particular CoS services. Figure 7-1 on page 152 displays the relationship of various CoS components to each other and illustrates the sequence in which they interact.

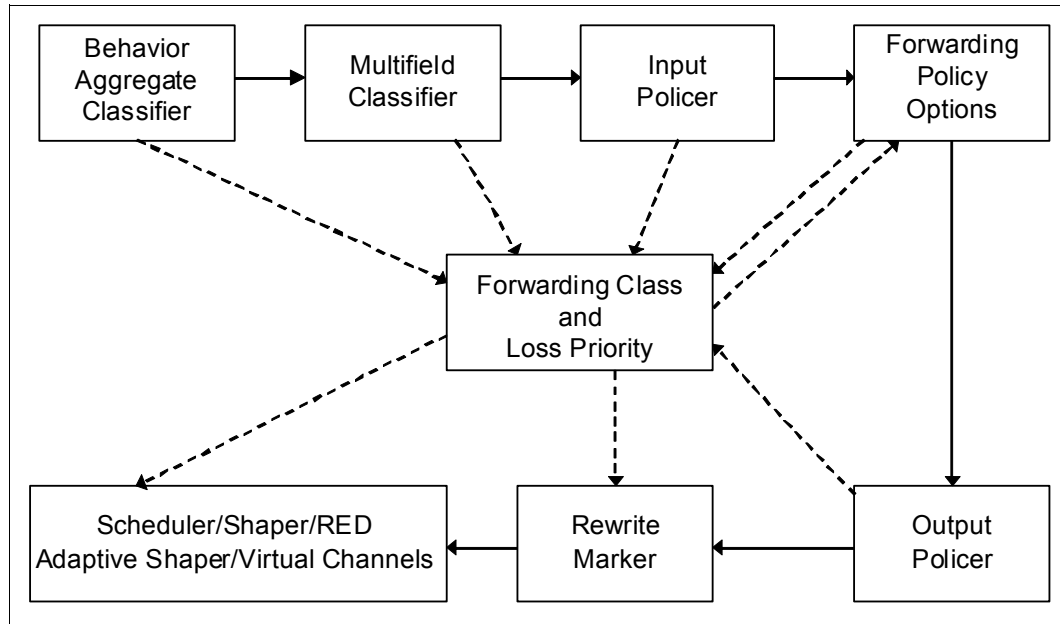


Figure 7-1 CoS packet flow

Typically, only a combination of some components is used to define a CoS service offering. For example, if a packet's class is determined by a behavior aggregate classifier, it is associated with a forwarding class and loss priority and does not need further classification by the multifield classifier.

CoS process on incoming packets

Classifiers and policers perform the following operations on incoming packets:

1. A classifier examines an incoming packet and assigns a forwarding class and loss priority to it.
2. Based on the forwarding class, the packet is assigned to an outbound transmission queue.
3. Input policers meter traffic to see if traffic flow exceeds its service level. Policers might discard, change the forwarding class and loss priority, or set the PLP bit of a packet. A packet for which the PLP bit is set has an increased probability of being dropped during congestion.

CoS process on outgoing packets

The scheduler map and rewrite rules perform the following operations on outgoing packets:

1. Scheduler maps are applied to interfaces and associate the outgoing packets with a scheduler and a forwarding class.
2. The scheduler defines how the packet is treated in the output transmission queue based on the configured transmit rate, buffer size, priority, and drop profile:
 - The buffer size defines the period for which the packet is stored during congestion.
 - The scheduling priority and transmit rate determine the order in which the packet is transmitted.
 - The drop profile defines how aggressively to drop packets that are using a particular scheduler.
3. Output policers meter traffic and might change the forwarding class and loss priority of a packet if a traffic flow exceeds its service level.

4. The rewrite rule writes information to the packet (for example, EXP, or DSCP bits) according to the forwarding class and loss priority of the packet.

7.4 Packet classification

Different QoS proposals have been standardized by the IETF. In particular, Junos software adheres to the differentiated services model (based on RFC 2475). Packets are normally classified at the edge of the network by marking the Differentiated Services code point (DSCP) field in the IP header, which essentially defines the class of service for that packet as it moves through the network.

As packets are classified in Junos software, they are assigned to a forwarding class that specifies both the transmit queue and the packet loss priority. The per-hop behavior of a packet is determined by its specific queue and loss priority. Later sections explain how traffic classes are specified and associated with queues, and how different loss priorities affect the drop probability of a packet.

There are two general types of classifiers:

- ▶ Behavior aggregate (BA) classifiers
- ▶ Multifield (MF) classifiers

For a specified interface, you can configure both an MF classifier and a BA classifier without conflicts. In such cases, BA classification is performed first, followed by MF classification. If the two classification results conflict, the MF classification result overrides the BA classification result.

7.4.1 Behavior aggregate classification

The behavior aggregate (BA) classifier maps a CoS value to a forwarding class and loss priority. The forwarding class determines the output queue. The loss priority is used by schedulers in conjunction with the random early discard (RED) algorithm to control packet discard during periods of congestion (see 7.10, “Red Profiles” on page 175).

The types of BA classifiers are based on which part of the incoming packet the classifier examines:

- ▶ Differentiated Services code point (DSCP) for IP DiffServ
- ▶ DSCP for IPv6 DiffServ
- ▶ IP precedence bits
- ▶ MPLS EXP bits
- ▶ IEEE 802.1p CoS bits
- ▶ IEEE 802.1ad drop eligible indicator (DEI) bit

Note:

- ▶ IEEE 802.1p BA classifier types are supported on devices operating in transparent mode.
- ▶ In Layer 2 mode, only IEEE 802.1p is supported.
- ▶ In Layer 3 mode, IPv4 Precedence, IPv4 DSCP, IPv6 DSCP are supported.

Default classification

The software automatically assigns an implicit default IP precedence classifier to all logical interfaces. If you enable the MPLS protocol family on a logical interface, a default MPLS EXP classifier is automatically applied to that logical interface.

Other default classifiers (such as those for IEEE 802.1p bits and DSCP) require that you explicitly associate a default classification table with a logical interface.

Table 7-1 shows the list of default classifiers.

Table 7-1 Default classifiers

Default classifier type	Default classifier name
Differentiated Services code point (DSCP)	dscp-default
Differentiated Services code point (DSCP) for IPv6	dscp-ipv6-default
MPLS experimental code point	exp-default
IEEE-802.1 code point	ieee8021p-default
IEEE-802.1ad (DEI) code point	ieee8021ad-default
IPv4 precedence code point	ipprec-default

You can see the default classifiers typing the command shown in the Example 7-1.

Example 7-1 DSCP default classifier

```
[edit]
ibm@J58S-2# run show class-of-service classifier name dscp-default
Classifier: dscp-default, Code point type: dscp, Index: 7
Code point      Forwarding class      Loss priority
000000          best-effort           low
000001          best-effort           low
000010          best-effort           low
000011          best-effort           low
000100          best-effort           low
000101          best-effort           low
000110          best-effort           low
000111          best-effort           low
001000          best-effort           low
001001          best-effort           low
001010          assured-forwarding    low
001011          best-effort           low
001100          assured-forwarding    high
001101          best-effort           low
001110          assured-forwarding    high
001111          best-effort           low
010000          best-effort           low
010001          best-effort           low
010010          best-effort           low
010011          best-effort           low
010100          best-effort           low
010101          best-effort           low
010110          best-effort           low
010111          best-effort           low
011000          best-effort           low
```

011001	best-effort	low
011010	best-effort	low
011011	best-effort	low
011100	best-effort	low
011101	best-effort	low
011110	best-effort	low
011111	best-effort	low
100000	best-effort	low
100001	best-effort	low
100010	best-effort	low
100011	best-effort	low
100100	best-effort	low
100101	best-effort	low
100110	best-effort	low
100111	best-effort	low
101000	best-effort	low
101001	best-effort	low
101010	best-effort	low
101011	best-effort	low
101100	best-effort	low
101101	best-effort	low
101110	expedited-forwarding	low
101111	best-effort	low
110000	network-control	low
110001	best-effort	low
110010	best-effort	low
110011	best-effort	low
110100	best-effort	low
110101	best-effort	low
110110	best-effort	low
110111	best-effort	low
111000	network-control	low
111001	best-effort	low
111010	best-effort	low
111011	best-effort	low
111100	best-effort	low
111101	best-effort	low
111110	best-effort	low
111111	best-effort	low

BA classifier configuration

You can override the default IP precedence classifier by defining a classifier and applying it to a logical interface. For each protocol family, classifiers are configured under [class-of-service], as shown in Example 7-2 on page 156.

A classifier takes a specified bit pattern and attempts to match it to the packet arriving on the interface. If the information in the packet's header matches the specified pattern, the packet is sent to the appropriate queue, which is defined by the forwarding class associated with the classifier.

Example 7-2 Classifier definition

```
[edit class-of-service]
  classifiers {
    (dscp | dscp-ipv6 | exp | ieee-802.1 | inet-precedence)
    classifier-name {
      import [classifier-name | default];
      forwarding-class class-name {
        loss-priority level {
          code-points [ aliases ] [
            bit-patterns ];
        }
      }
    }
  }
}
```

For a complete description of the command type:

```
ibm@J58S-2> help topic class-of-service classifiers
```

You can apply the classifier to an interface as shown in Example 7-3.

Example 7-3 Applying a classifier

```
class-of-service {
  interface interface-name {
    unit unit-number {
      classifiers {
        dscp|dscp-ipv6|exp|inet-precedence classifier-name;
      }
    }
  }
}
```

BA classifier example

In Example 7-4, we define an IP Precedence Classifier named TEST-CLASSIFIER and associate code points: 001, 010, and 011 for wading class Assured Forwarding with low loss priority. Then, we verify the configuration and apply the classifier to interface ge-0/0/0.

Example 7-4 Classifier configuration

```
[edit class-of-service]
ibm@J58S-2# set classifiers inet-precedence TEST-CLASSIFIER forwarding-class
assured-forwarding loss-priority low code-points 001 code-points 010 code-points
011
```

```
[edit class-of-service]
ibm@J58S-2# commit
commit complete
```

```
[edit class-of-service]
ibm@J58S-2# run show class-of-service classifier name TEST-CLASSIFIER
Classifier: TEST-CLASSIFIER, Code point type: inet-precedence, Index: 28143
  Code point      Forwarding class      Loss priority
  001             assured-forwarding      low
  010             assured-forwarding      low
  011             assured-forwarding      low
```



```
[edit class-of-service]
ibm@J58S-2# set interfaces ge-1/0/9 unit 0 classifiers inet-precedence
TEST-CLASSIFIER

[edit class-of-service]
ibm@J58S-2# commit

[edit class-of-service]
ibm@J58S-2# run show class-of-service interface ge-1/0/9
Physical interface: ge-1/0/9, Index: 141
Queues supported: 4, Queues in use: 4
  Scheduler map: <default>, Index: 2
  Chassis scheduler map: <default-chassis>, Index: 4

Logical interface: ge-1/0/9.0, Index: 70
  Object      Name      Type      Index
  Classifier   TEST-CLASSIFIER   ip      28143
```

7.4.2 Code point alias

Behavior aggregate (BA) classifiers use class-of-service (CoS) values, such as Differentiated Services code points (DSCPs), DSCP IPv6, IP precedence, IEEE 802.1 and MPLS experimental (EXP) bits to associate incoming packets with a particular CoS servicing level. You can assign a meaningful name or alias to the CoS values and use this alias instead of bits when configuring CoS components. These aliases are not part of the specifications but are well known through usage, for example, the alias for DSCP 101110 is widely accepted as Expedited Forwarding (EF).

When you configure classes and define classifiers, you can refer to the markers by alias names. You can configure user-defined classifiers in terms of alias names. If the value of an alias changes, it alters the behavior of any classifier that references it.

Default aliases

Example 7-5 shows the default mappings between DSCP bit values and standard aliases. You can see the default mappings for others bit codes (IP Precedence, IEEE 802.1p, MPLS EXP, and so on) issuing the same command.

Example 7-5 DSCP code point alias

```
{primary:node0}
ibm@J58S-1> show class-of-service code-point-aliases dscp
Code point type: dscp
Alias      Bit pattern
af11      001010
af12      001100
af13      001110
af21      010010
af22      010100
af23      010110
af31      011010
af32      011100
af33      011110
```

af41	100010
af42	100100
af43	100110
be	000000
cs1	001000
cs2	010000
cs3	011000
cs4	100000
cs5	101000
cs6	110000
cs7	111000
ef	101110
nc1	110000
nc2	111000

Code point alias configuration

To configure CoS code point aliases, include the `code-point-aliases` statement at the [edit class-of-service] hierarchy level, as shown in Example 7-6.

Example 7-6 Defining code point aliases

```
code-point-aliases {
  (dscp | dscp-ipv6 | exp | ieee-802.1 | inet-precedence) {
    alias-name bits;
  }
}
```

Code point alias example

In Example 7-7, we generate an alias named `ea` for DSCP code point 001010.

Example 7-7 Configuring code point alias

```
{primary:node0}[edit]
ibm@J58S-1# set class-of-service code-point-aliases dscp ea 001010

{primary:node0}[edit]
ibm@J58S-1# commit

{primary:node0}[edit]
ibm@J58S-1# run show class-of-service code-point-aliases dscp
Code point type: dscp
Alias          Bit pattern
af11           001010
af12           001100
af13           001110
af21           010010
af22           010100
af23           010110
af31           011010
af32           011100
af33           011110
af41           100010
af42           100100
af43           100110
```

be	000000
cs1	001000
cs2	010000
cs3	011000
cs4	100000
cs5	101000
cs6	110000
cs7	111000
ea	001010
ef	101110
nc1	110000
nc2	111000

7.4.3 Multifield classification

A Multifield Classifier (MF) provides the filtering functionality that scans through a variety of packet fields to determine the forwarding class for a packet. Typically, a classifier performs matching operations on the selected fields against a configured value.

A multifield classifier typically matches one or more of the six packet header fields:

- ▶ Destination address
- ▶ Source address
- ▶ IP protocol
- ▶ Source port
- ▶ Destination port
- ▶ DSCP

Multifield classifiers are used when a simple BA classifier is insufficient to classify a packet.

Configuration

The configuration of an MF classifier requires the definition of a firewall filter, which is located under the [firewall family inet] hierarchy as shown in Example 7-8. Be aware that 802.1p bits cannot be used as matching conditions; by the time the packet is processed by the ingress/egress filters, the Ethernet frame has been stripped.

Example 7-8 Firewall filter definition for MF configuration

```
firewall {
    family inet{
        filter filter-name {
            term term-name {
                from {
                    match-conditions;
                }
                then {
                    forwarding-class class-name;
                    loss-priority priority;
                }
            }
        }
    }
}
```

For a complete description of the command type:

```
ibm@J58S-2# help topic firewall filter
```

Filters can be applied to logical interfaces both on ingress and egress, as we show in Example 7-9.

Example 7-9 Applying filters

```
interfaces interface-name{
    unit unit-number {
        family inet {
            filter {
                input filter-name;
                output filter-name;
            }
        }
    }
}
```

Example

In Example 7-10, we classify traffic with source address 10.0.0.0/24 and assign it to forwarding class Assured Forwarding with medium-low loss priority. Then we apply the filter to interface ge-1/0/9.

Example 7-10 Multifield configuration

```
[edit]
ibm@J58S-2# set firewall family inet filter CoS-TEST term SOURCE from
source-address 10.0.0.0/24

[edit]
ibm@J58S-2# set firewall family inet filter CoS-TEST term SOURCE then
forwarding-class assured-forwarding

[edit]
ibm@J58S-2# set firewall family inet filter CoS-TEST term SOURCE then
loss-priority medium-low

[edit]
ibm@J58S-2# set interfaces ge-1/0/9 unit 0 family inet filter input CoS-TEST

[edit]
ibm@J58S-2# show firewall family inet filter CoS-TEST
term SOURCE {
    from {
        source-address {
            10.0.0.0/24;
        }
    }
    then {
        loss-priority medium-low;
        forwarding-class assured-forwarding;
    }
}
```

```
[edit]
ibm@J58S-2# show interfaces ge-1/0/9
unit 0 {
    family inet {
        filter {
            input CoS-TEST;
        }
    }
}
```

7.5 Forwarding classes

It is helpful to think of forwarding classes as output queues. In effect, the end result of classification is the identification of an output queue for a particular packet. IBM j-type s-series supports eight queues (0 through 7). For a classifier to assign an output queue to each packet, it must associate the packet with one of the following forwarding classes:

- ▶ Expedited forwarding (EF): Provides a low loss, low latency, low jitter, assured bandwidth, end-to-end service.
- ▶ Assured forwarding (AF): Provides a group of values you can define and includes four subclasses: AF1, AF2, AF3, and AF4, each with three drop probabilities: low, medium, and high.
- ▶ Best effort (BE): Provides no service profile. For the BE forwarding class, loss priority is typically not carried in a class-of-service (CoS) value, and random early detection (RED) drop profiles are more aggressive.
- ▶ Network Control (NC): This class is typically high priority because it supports protocol control.

Default forwarding classes

By default, four queues are assigned to four forwarding classes, each with a queue number, name, and abbreviation.

The four forwarding classes defined by default are shown in Table 7-2. If desired, you can rename the forwarding classes associated with the queues supported on your hardware.

Table 7-2 Default forwarding classes

Queue	Forwarding class name	Description
Queue 0	best-effort (be)	The software does not apply any special CoS handling to packets with 000000 in the DiffServ field, a backward compatibility feature. These packets are usually dropped under congested network conditions.
Queue 1	expedited-forwarding (ef)	The software delivers assured bandwidth, low loss, low delay, and low delay variation (jitter) end-to-end for packets in this service class. Routers accept excess traffic in this class, but in contrast to assured forwarding, out-of-profile expedited-forwarding packets can be forwarded out of sequence or dropped.

Queue	Forwarding class name	Description
Queue 2	assured-forwarding (af)	The software offers a high level of assurance that the packets are delivered as long as the packet flow from the customer stays within a certain service profile that you define. The software accepts excess traffic, but applies a RED drop profile to determine if the excess packets are dropped and not forwarded. Depending on router type, up to four drop probabilities (low, medium-low, medium-high, and high) are defined for this service class.
Queue 3	network-control (nc)	The software delivers packets in this service class with a low priority. (These packets are not delay sensitive.) Typically, these packets represent routing protocol hello or keepalive messages. Because loss of these packets jeopardizes proper network operation, delay is preferable to discard.

Forwarding classes configuration

You assign each forwarding class to an internal queue number by including the forwarding-classes statement at the [edit class-of-service] hierarchy level, as shown in Example 7-11.

Example 7-11 Forwarding class definition

```
[edit class-of-service]
forwarding-classes {
    queue queue-number class-name;
}
```

In addition to BA and MF classification, the forwarding class of a packet can be directly determined by the logical interface that receives the packet. This FC of a packet can be configured using CLI commands, as shown in Example 7-12, and if configured, this FC overrides the FC from any BA classification that was previously performed on the logical interface.

Example 7-12 Direct forwarding class assignment

```
class-of-service {
    interfaces {
        interface-name {
            unit unit-number {
                forwarding-class class-name;
            }
        }
    }
}
```

7.6 Simple filters and policers

To handle oversubscribed traffic in the IBM j-type Ethernet appliances, you can configure simple filters and policing.

Filter or policing actions: For IBM Ethernet Appliance J56S and J58S devices, the simple filter or policing actions can be applied only to logical interfaces residing in an line Flex IOC (FIOC) because only a line FIOC supports the simple filter and policing features on the IBM Ethernet Appliance J56S and J58S devices.

7.6.1 Simple filters

The simple filter functionality comprises of the following:

- ▶ Classifying packets according to configured policies
- ▶ Taking appropriate actions based on the results of classification

In Junos software, ingress traffic policers can limit the rate of incoming traffic. There are two main reasons to use traffic policing:

- ▶ To enforce traffic rates to conform to the service-level agreement (SLA)
- ▶ To protect next hops, for example, protecting the central point and the SPU from being overwhelmed by excess traffic (example, DOS attacks)

Using the results of packet classification and traffic metering, a policer can take one of the following actions for a packet: forward a conforming (green) packet or drop a nonconforming (yellow) packet. Policers always discard a non conforming red packet. The traffic metering supports the algorithm of the two-rate tricolor marker (TCM) (RFC 2698).

Configuring a Simple Filter

Simple filters, in contrast to other firewall filters, support only a subset of the full firewall filter syntax. Unlike normal filters, simple filters are for IPv4 traffic only and have the following restrictions:

- ▶ The next term action is not supported.
- ▶ Qualifiers, such as the except and protocol-except statements, are not supported.
- ▶ Noncontiguous masks are not supported.
- ▶ Multiple source addresses and destination addresses in a single term are not supported. If you configure multiple addresses, only the last one is used.
- ▶ Ranges are not supported.
- ▶ Output filters are not supported. You can apply a simple filter to inbound (ingress) traffic only.

To configure a simple filter, include the statement from Example 7-13 at the *[edit firewall]* hierarchy level of the configuration.

Example 7-13 Simple filter

```
firewall {
  family inet {
    simple-filter sf-1 {
      term 1 {
        source-address 172.16.0.0/16;
        destination-address 20.16.0.0/16;
        source-port 1024;
      }
      then { # Action with term-1
        forwarding-class fc-bel;
        loss-priority high;
      }
    }
  }
}
```

```

    }
    term 2 {
        source-address 173.16.0.0/16;
        destination-address 21.16.0.0/16;
    }
    then { # Action with term-2
        forwarding-class fc-ef1;
        loss-priority low;
    }
}
interfaces { # Apply the simple filter.
ge-1/2/3 {
    unit 0 {
        family inet {
            simple-filter {
                input sf-1;
            }
        }
    }
}
}

```

Applying a simple filter

A simple filter can be applied to logical interfaces. Use the Example 7-14 CLI commands to apply a simple filter

Example 7-14 Applying a simple filter

```

edit interfaces interface-name unit logical-unit-number family family-name
simple-filter {
    input filter-name;
}

```

Note: You can apply simple filters to the family inet only, and only in the input direction. Because of hardware limitations on the IBM j-type Ethernet Appliances, a maximum of 400 logical input interfaces (in one broadcom packet processor) can be applied with simple filters.

Applying a policer through a simple filter

In Junos software, policers can be configured as part of the firewall filter hierarchy. You can configure a policer and then apply it as one of the actions of a term in a simple filter. The policer can limit the rate of traffic that enters the logical interface to which the simple filter is applied. Figure 7-2 on page 165 illustrates the application of a policer.

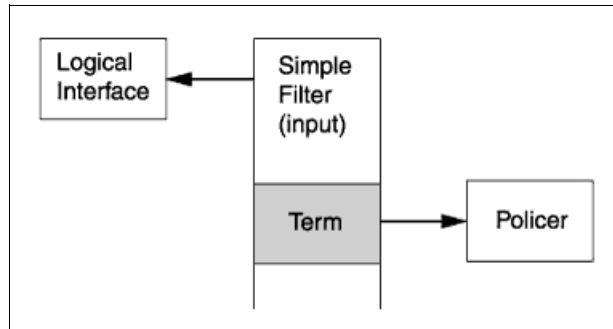


Figure 7-2 Application of a policer through a simple filter

7.6.2 Policing

Policing refers to the ability of a router to measure data rates and, based on this measurement, to either drop or reclassify the traffic.

After Multifield Classification is performed, it is possible to instruct a router to measure the rate of the traffic matching the classifier, and either drop or change the forwarding class, or drop the priority of the packet if the measured rate exceeds a configurable threshold.

In simple terms, policers allow the establishment of a data rate, which, if exceeded, results in traffic being either reclassified or dropped. In order to measure traffic rates, it is important to determine a measurement interval (or burst limits). Traffic always egresses an interface at line rate. To send traffic at a “lower speed,” bursts have to be followed by idle periods, resulting in an average transmit rate lower than the line rate.

There are three types of policers:

- Two-color marking

A two-color policer meters the traffic stream and classifies packets into two categories of packet loss priority (PLP) according to a configured bandwidth and burst-size limit. You can mark packets that exceed the bandwidth and burst-size limit in some way, or simply discard them.

- Single-rate tricolor marking

This type of policer, defined in RFC 2697, meters traffic based on the configured Committed Information Rate (CIR), Committed Burst Size (CBS), and the Excess Burst Size (EBS). Traffic is marked as belonging to one of three categories (green, yellow, or red) based on whether the packets arriving are below the CBS (green), exceed the CBS (yellow) but not the EBS, or exceed the EBS (red).

- Two-rate tricolor marking

This type of policer, defined in RFC 2698, meters traffic based on the configured CIR and Peak Information Rate (PIR), along with their associated burst sizes, the CBS and peak burst size (PBS). Traffic is marked as belonging to one of three categories (green, yellow, or red) based on whether the packets arriving are below the CIR (green), exceed the CIR (yellow) but not the PIR, or exceed the PIR (red).

7.6.3 Policing configuration (Two-color marking)

Besides the classification mentioned in the previous section, you can configure two types of policers:

- Filter based policers: The policer is applied in the firewall filter as an action.
- Interface based policers: The policer is applied directly in the logical interface.

Important: IBM j-type s-series J56S and J58S allow only inbound (ingress) interface policers.

For a complete list of filter and policing limitations on Junos 10.1 for IBM j-type s-series refer to Juniper Networks JUNOS 10.1 Software Release Notes available at:

http://www.juniper.net/techpubs/en_US/junos10.1/information-products/topic-collections/release-notes/10.1/junos-release-notes-10.1.pdf

Filter-based policers

The configuration required uses firewall filters similar to those used for packet classification. The first step is to define a policer under the [firewall policers] hierarchy, as shown in Example 7-15.

Example 7-15 Firewall policer definition

```
[edit firewall]
  policer policer-name {
    if-exceeding {
      bandwidth-limit bps;
      bandwidth-percent number;
      burst-size-limit bytes;
    }
    then {
      discard;
    }
  }
```

For a complete description of the command type:

```
ibm@J11M-re0# help topic firewall policer
```

A policer specifies a maximum bandwidth, a max-burst size, and an action. The suggested policer's burst limit value for a low-speed interface is ten times the interface MTU. For a high-speed interface, the suggested burst size is the transmit rate of the interface times 3 to 5 milliseconds.

Filter-based policers are applied as an action in a firewall filter, as shown in Example 7-16.

Example 7-16 Policer applied to a firewall filter

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

```

```

        policer policer-name;
        other actions;
    }
}
}
}
}

```

Interface-based policer

You can use policers to limit the amount of traffic passing into or out of an interface. Consider the following information when applying a policer to an ingress interface:

- ▶ You can configure a tricolor marker in the firewall; however, you cannot use the marker in the input policer.
- ▶ Only the following options are valid: **logical-interface-policer**, **if-exceeding**, and **then**.
- ▶ For the **if-exceeding** option, only **bandwidth-limit** and **burst-size-limit** are valid options. The **bandwidth-percent** option is not supported.
- ▶ For the **then** option, only **discard** is the valid option.

To configure an interface-based policer, first define a policer, as show in Example 7-15 on page 166. Then apply the policer to the logical interface, as shown in Example 7-17.

Example 7-17 Policer applied to an interface

```

[edit interfaces]
interface-name{
    unit unit-name{
        family inet {
            policer {
                input policer-name;
            }
        }
    }
}

```

7.6.4 Policing example (Interface-based policer)

In Example 7-18, we show the configuration of a policer, named TEST-POLICER, which limits the input traffic to 2 Mbps on interface ge-1/0/9.0 on an IBM j-type s-series.

Example 7-18 Interface-based policer configuration

```

[edit]
ibm@J58S-2# set firewall policer TEST-POLICER if-exceeding bandwidth-limit 2000000
burst-size-limit 375k

[edit]
ibm@J58S-2# set firewall policer TEST-POLICER then discard

[edit]
ibm@J58S-2# show firewall policer TEST-POLICER
if-exceeding {
    bandwidth-limit 2m;
    burst-size-limit 375k;
}

```

```

then discard;

ibm@J58S-2# set interfaces ge-1/0/9 unit 0 family inet policer input TEST-POLICER

[edit]
ibm@J58S-2# show interfaces ge-1/0/9
unit 0 {
    family inet {
        policer {
            output TEST-POLICER;
        }
    }
}

[edit]
ibm@J58S-2# commit

[edit]
ibm@J58S-2# run show policer TEST-POLICER-ge-1/0/9.0-inet-i
Policers:
Name                                     Packets
TEST-POLICER-ge-1/0/9.0-inet-i         0

```

7.7 Rewrite rules

As packets enter or exit a network, edge routers might be required to alter the class-of-service (CoS) settings of the packets. Rewrite rules set the value of the CoS bits within the packet's header. Each rewrite rule reads the current forwarding class and loss priority information associated with the packet, locates the chosen CoS value from a table, and writes this CoS value into the packet header.

You configure rewrite rules to alter CoS values in outgoing packets on the outbound interfaces of an edge router to meet the policies of a targeted peer. This allows the downstream router in a neighboring network to classify each packet into the appropriate service group.

In addition, you often need to rewrite a given marker (IP precedence, Differentiated Services code point [DSCP], IEEE 802.1p, or MPLS EXP settings) at the inbound interfaces of an edge router to accommodate Behavior Aggregate classification by core devices.

7.7.1 Default rewrite rules

By default, rewrite rules are not usually applied to interfaces. If you want to apply a rewrite rule, you can either design your own rule and apply it to an interface, or you can apply a default rewrite rule. Table 7-3 shows the defaults rewrite rules tables. To see the contents of these tables use the command shown in Example 7-19 on page 169.

Table 7-3 Default rewrite rules

Default rewrite rule type	Default rewrite rule Name
Differentiated Services code point (DSCP)	dscp-default
Differentiated Services code point (DSCP) for IPv6	dscp-ipv6-default

Default rewrite rule type	Default rewrite rule Name
MPLS experimental code point	exp-default
IEEE-802.1 code point	ieee8021p-default
IEEE-802.1ad (DEI) code point	ieee8021ad-default
IPv4 precedence code point	ipprec-default

Example 7-19 Dscp-default rewrite rule

```
[edit]
ibm@J58S-1# run show class-of-service rewrite-rule name dscp-default
Rewrite rule: dscp-default, Code point type: dscp, Index: 31
  Forwarding class      Loss priority  Code point
  best-effort           low          000000
  best-effort           high         000000
  expedited-forwarding low          101110
  expedited-forwarding high         101110
  assured-forwarding   low          001010
  assured-forwarding   high         001100
  network-control      low          110000
  network-control      high         111000
```

7.7.2 Rewrite rules configuration

To configure a rewrite rule mapping and associate it with the appropriate forwarding class and code-point alias or bit set, include the **rewrite-rules** statement at the [edit class-of-service] hierarchy level, as shown in Example 7-20.

Example 7-20 Configuring rewrite rules

```
[edit class-of-service]
  rewrite-rules {
    (dscp | dscp-ipv6 | exp | ieee-802.1 | inet-precedence)
    rewrite-name {
      import (rewrite-name | default);
      forwarding-class class-name {
        loss-priority level code-point (alias |
        bits);
      }
    }
  }
```

For a complete description of the command type:

```
ibm@J11M-re0# help topic class-of-service rewrite-rules
```

Example 7-21 shows how to apply a rewrite rule to a logical interface.

Example 7-21 Applying rewrite rules to logical interfaces

```
[edit class-of-service interfaces interface-name unit logical unit-number]
  rewrite-rules {
    dscp (rewrite-name | default) protocol protocol-types;
```

```

ieee-802.1 (rewrite-name | default) inet-prec vlan-tag (outer |
outer-and-inner);
inet-precedence (rewrite-name | default) protocol protocol-types;
}

```

7.7.3 Rewrite rules example

In Example 7-22, we show how to apply the default **dscp** rewrite rule to interface ge-1/0/9 unit 0.

Example 7-22 Applying dscp default rewrite rule

```

[edit]
ibm@J58S-2# set class-of-service interfaces ge-1/0/9 unit 0 rewrite-rules dscp
default

```

```

[edit]
ibm@J58S-2# commit

```

```

[edit]
ibm@J58S-2# show class-of-service interfaces ge-1/0/9
unit 0 {
    rewrite-rules {
        dscp default;
    }
}

```

```

[edit]
ibm@J58S-2# run show class-of-service interface ge-1/0/9
Physical interface: ge-1/0/9, Index: 141
Queues supported: 4, Queues in use: 4
Scheduler map: <default>, Index: 2
Chassis scheduler map: <default-chassis>, Index: 4

```

Logical interface: ge-1/0/9.0, Index: 70

Object	Name	Type	Index
Rewrite	dscp-default	dscp	31
Classifier	TEST-CLASSIFIER	ip	28143

7.8 Packet's Loss Priority (PLP)

Junos software permits the tagging of packets with a Loss Priority indicator. This tag performs similar functions as the Discard Eligibility bit in frame relay or Cell Loss Priority bit in ATM.

Internally, PLP bits cannot be viewed directly in the device. Externally, PLP bits are carried using specific bits in the IP precedence, MPLS EXP or DSCP fields. Default DSCP and IP precedence rewrite and classification tables do not support external communication of PLP for the BE and EF classes.

You can use the loss priority setting to identify packets that have experienced congestion. Typically you mark packets exceeding some service level with a high loss priority. You set loss priority by configuring a classifier or a policer. The loss priority is used later in the work flow to select one of the drop profiles used by RED, as shown in 7.10, "Red Profiles" on page 175.

For more information about this topic refer to section Packet Loss Priority Configuration Overview on *Junos software Class of Service Configuration Guide*:

http://www.juniper.net/techpubs/en_US/junos10.1/information-products/topic-collections/config-guide-cos/config-guide-cos.pdf

7.9 Schedulers

After traffic is classified, queuing and scheduling can be used to provide different levels of service for the classified traffic. In Junos software, forwarding classes are associated with queues.

You use schedulers to define the properties of output queues. These properties include the amount of interface bandwidth assigned to the queue, the size of the memory buffer allocated for storing packets, the priority of the queue, and the random early detection (RED) drop profiles associated with the queue. You associate the schedulers with forwarding classes by means of scheduler maps. You can then associate each scheduler map with an interface, thereby configuring the hardware queues, packet schedulers, and RED processes that operate according to this mapping.

7.9.1 Default schedulers

Only two forwarding classes (queues) have default scheduler configuration:

- ▶ Best Effort (queue 0): Receives 95 percent of the bandwidth
- ▶ Network Control (queue 3): Receives 5 percent of the bandwidth.

The default drop profile causes the buffer to fill and then discard all packets until it has space.

By default the expedited forwarding and assured forwarding classes have no schedulers. You can manually configure resource for these classes.

Also by default, each queue can exceed the assigned bandwidth if additional bandwidth is available.

7.9.2 Schedulers' configuration

The following terms describe the parameters needed to configure schedulers.

▶ Transmission Rate

The transmission-rate control determines the actual traffic bandwidth from each forwarding class you configure. The rate is specified in bits per second. Each queue is allocated some portion of the bandwidth of the outgoing interface.

This bandwidth amount can be a fixed value, such as 1 megabit per second (Mbps), a percentage of the total available bandwidth, or the rest of the available bandwidth. You can allow transmission bandwidth to exceed the configured rate if additional bandwidth is available from other queues. In case of congestion, configured amount of transmission rate is guaranteed for the queue. This property allows you to ensure that each queue receives the amount of bandwidth appropriate to its level of service.

▶ Scheduler Buffer Size

To control congestion at the output stage, you can configure the delay-buffer bandwidth. The delay-buffer bandwidth provides packet buffer space to absorb burst traffic up to the

specified duration of delay. After the specified delay buffer becomes full, packets with 100 percent drop probability are dropped from the tail of the buffer.

For each scheduler, you can configure the buffer size as one of the following:

- A percentage of the total buffer.
- The remaining buffer available. The remainder is the buffer percentage that is not assigned to other queues.

► Shaping rate

Shaping rates control the maximum rate of traffic transmitted on an interface. You can configure the shaping rate so that the interface transmits less traffic than it is physically capable of carrying.

You can configure shaping rates on logical interfaces. By default, output scheduling is not enabled on logical interfaces. Logical interface scheduling (also called per-unit scheduling) allows you to enable multiple output queues on a logical interface and associate an output scheduler and shaping rate with the queues.

► Priority scheduling

Priority scheduling determines the order in which an output interface transmits traffic from the queues, thus ensuring that queues containing important traffic are provided better access to the outgoing interface.

Priority scheduling is accomplished through a procedure in which the scheduler examines the priority of the queue. Higher-priority queues transmit packets ahead of lower priority queues as long as the higher-priority forwarding classes retain enough bandwidth credit.

IBM j-type s-series support the following priority levels:

- Low
- Medium-low
- Medium-high
- High
- Strict-high

► Scheduler drop profile maps

Drop-profile maps associate drop profiles with a scheduler. Drop-profile map sets the drop profile for a specific packet loss priority (PLP) and protocol type. The inputs for the drop-profile map are the PLP and the protocol type. The output is the drop profile. The scheduler drop profile defines the drop probability for the Random Early Detection (RED) process.

To configure schedulers:

1. Create a scheduler template, as shown in Example 7-23 on page 173.
2. Define a scheduler map that associates each scheduler with a forwarding class. See Example 7-24 on page 173.
3. Apply the scheduler map to the interface, as shown in Example 7-25 on page 173.
Generally, you can associate schedulers with physical interfaces only (for some interfaces, you can also associate schedulers with the logical interface).

Example 7-23 Scheduler template definition

```
class-of-service {
    schedulers name {
        priority strict-high | high | medium-high | medium-low | low;
        transmit-rate (rate | percent percentage | remainder);
    }
}
```

Example 7-24 Scheduler map definition

```
edit class-of-service {
    scheduler-maps {
        map-name {
            forwarding-class class-name scheduler scheduler-name;
        }
    }
}
```

Example 7-25 Applying a scheduler map to an interface

```
class-of-service {
    interfaces interface-name {
        scheduler-map scheduler-map-name;
    }
}
```

7.9.3 Schedulers' example

In Example 7-26, we begin the configuration to define the scheduler named TEST-SCHEDULER and to assign 20 percent of the bandwidth. In addition, low priority is configured.

Next, we define a scheduler map and associate the previous scheduler with the best-effort forwarding class.

Finally, we apply the scheduler map to interface ge-1/0/9.

Example 7-26 Scheduler configuration

```
[edit]
ibm@J58S-2# set class-of-service schedulers TEST-SCHEDULER transmit-rate percent
20 exact

[edit]
ibm@J58S-2# set class-of-service schedulers TEST-SCHEDULER priority low

[edit]
ibm@J58S-2# show class-of-service schedulers
TEST-SCHEDULER {
    transmit-rate percent 20;
    priority low;
}

[edit]
```

```
ibm@J58S-2# set class-of-service scheduler-maps TEST-SCHEDULER-MAP
forwarding-class best-effort scheduler TEST-SCHEDULER
```

```
[edit]
ibm@J58S-2# set class-of-service interfaces ge-1/0/9 scheduler-map
TEST-SCHEDULER-MAP
```

```
[edit]
ibm@J58S-2# show class-of-service interfaces
ge-1/0/9 {
    scheduler-map TEST-SCHEDULER-MAP;
    unit 0 {
        classifiers {
            inet-precedence TEST-CLASSIFIER;
        }
        rewrite-rules {
            dscp default;
        }
    }
}
```

```
[edit]
ibm@J58S-2# commit
```

```
[edit]
ibm@J58S-2# run show class-of-service interface ge-1/0/9
Physical interface: ge-1/0/9, Index: 141
Queues supported: 4, Queues in use: 4
Scheduler map: TEST-SCHEDULER-MAP, Index: 17100
Chassis scheduler map: <default-chassis>, Index: 4
```

```
Logical interface: ge-1/0/9.0, Index: 70
```

Object	Name	Type	Index
Rewrite	dscp-default	dscp	31
Classifier	TEST-CLASSIFIER	ip	28143

```
[edit]
ibm@J58S-2# run show class-of-service scheduler-map TEST-SCHEDULER-MAP
Scheduler map: TEST-SCHEDULER-MAP, Index: 17100
```

```
Scheduler: TEST-SCHEDULER, Forwarding class: best-effort, Index: 49130
Transmit rate: 20 percent, Rate Limit: none, Buffer size: remainder, Buffer
Limit: none, Priority: low
Excess Priority: unspecified
Drop profiles:
```

Loss priority	Protocol	Index	Name
Low	any	1	<default-drop-profile>
Medium low	any	1	<default-drop-profile>
Medium high	any	1	<default-drop-profile>
High	any	1	<default-drop-profile>

7.10 Red Profiles

Junos software implements the Random Early Detection (RED) algorithm, which was designed to prevent TCP synchronization when links experience congestion. When data networks become congested and drop packets, TCP sessions that suffer packet loss will reduce their window size to avoid congestion. Indiscriminate packet drops will (statistically speaking) signal the transmitting endpoints to slow their transmission rates, causing most of the senders to exponentially decrease their transmission rates simultaneously. This phenomenon is known as TCP synchronization, and it leads to bandwidth fluctuations on congested links. When synchronization occurs, senders reduce their transmission rates simultaneously, and slowly increase them again until links become congested, a process that then repeats itself.

The random early detection (RED) algorithm solves this by randomly dropping packets as queues become full. The drop probability can be configured as a function of queue size at any given time, so the more congestion, the more aggressive the drop profile. Randomly dropping traffic before an interface becomes congested signals end hosts to slow down, preventing an overloaded queue.

You configure RED using drop profiles, which is a mechanism that defines parameters that allow packets to be dropped from the network. Drop profiles define the meanings of the loss priorities.

When you configure drop profiles, there are two important values: the *queue fullness* and the *drop probability*. The queue fullness represents a percentage of the memory used to store packets in relation to the total amount that has been allocated for that specific queue. Similarly, the drop probability is a percentage value that correlates to the likelihood that an individual packet is dropped from the network.

7.10.1 RED Profile configuration

Loss profiles can be specified in two ways:

- ▶ Staircase type profile
You have to specifying a set of fill levels with associated drop probabilities. The drop probability is assumed to be constant between fill levels.
- ▶ Interpolated profile
You can create piecewise linear functions to specify the drop probability.

Loss profiles are configured under the [class-of-service drop-profiles] hierarchy, as shown in Example 7-27.

Example 7-27 RED drop profile definition

```
[edit class-of-service]
  drop-profiles {
    profile-name {
      fill-level percentage drop-probability percentage;
      interpolate {
        drop-probability [ values ];
        fill-level [ values ];
      }
    }
  }
```

After configuring the drop profiles, they must be applied to a scheduler, as shown in Example 7-28. For more information about schedulers refer to 7.9, “Schedulers” on page 171.

Example 7-28 Applying drop profile to scheduler

```
class-of-service {
  schedulers {
    drop-profile-name {
      drop-profile-map loss-priority priority protocol any drop-profile
name;
    }
  }
}
```

7.10.2 RED Profile example

Example 7-29 shows how to generate the staircase profile specified in Figure 7-3 on page 177.

Example 7-30 on page 177 shows how to generate the interpolated profile specified in Figure 7-4 on page 179

Example 7-29 Staircase RED profile configuration

```
[edit]
ibm@J58S-2# set class-of-service drop-profiles STAIR-TEST fill-level 70
drop-probability 10

[edit]
ibm@J58S-2# set class-of-service drop-profiles STAIR-TEST fill-level 80
drop-probability 20

[edit]
ibm@J58S-2# set class-of-service drop-profiles STAIR-TEST fill-level 90
drop-probability 25

ibm@J58S-2# show class-of-service drop-profiles
STAIR-TEST {
    fill-level 70 drop-probability 10;
    fill-level 80 drop-probability 20;
    fill-level 90 drop-probability 25;
}

[edit]
ibm@J58S-2# commit

[edit]
ibm@J58S-2# run show class-of-service drop-profile STAIR-TEST
Drop profile: STAIR-TEST, Type: discrete, Index: 52959
  Fill level    Drop probability
      70         10
      80         20
      90         25
```

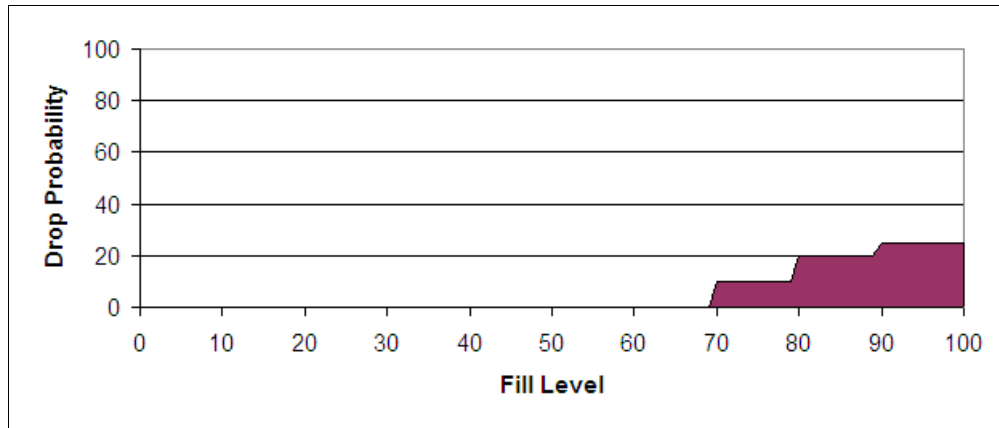


Figure 7-3 Staircase RED profile

Example 7-30 Interpolated RED profile configuration

```
[edit]
ibm@J58S-2# set class-of-service drop-profiles INTERPOLATE-TEST interpolate
fill-level [60 70 80 90 ] drop-probability [0 15 35 ]
```

```
[edit]
ibm@J58S-2# show class-of-service drop-profiles
INTERPOLATE-TEST {
    interpolate {
        fill-level [ 60 70 80 90 ];
        drop-probability [ 0 5 15 35 ];
    }
}
```

```
[edit]
ibm@J58S-2# commit
```

```
[edit]
ibm@J58S-2# run show class-of-service drop-profile
Drop profile: <default-drop-profile>, Type: discrete, Index: 1
  Fill level    Drop probability
    100         100
Drop profile: INTERPOLATE-TEST, Type: interpolated, Index: 7093
  Fill level    Drop probability
    0           0
    1           0
    2           0
    4           0
    5           0
    6           0
    8           0
   10           0
   12           0
   14           0
   15           0
   16           0
   18           0
   20           0
```

22	0
24	0
25	0
26	0
28	0
30	0
32	0
34	0
35	0
36	0
38	0
40	0
42	0
44	0
45	0
46	0
48	0
49	0
51	0
52	0
54	0
55	0
56	0
58	0
60	0
62	1
64	2
65	2
66	3
68	4
70	5
72	7
74	9
75	10
76	11
78	13
80	15
82	19
84	23
85	25
86	27
88	31
90	35
92	48
94	61
95	67
96	74
98	87
99	93
100	100

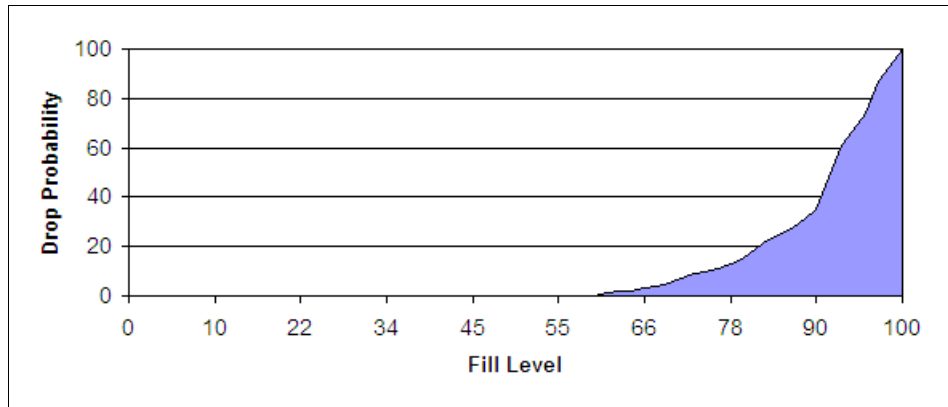


Figure 7-4 Interpolated RED profile

7.11 Case study

The purpose of this section is to show the configuration of the following CoS features in the scenario depicted in Figure 7-5 on page 180.

- **Multifield Classification**

HTTP traffic from J02M will be classified at the ingress in J58S and assigned to Expedited Forwarding forwarding class.

- **Policing**

All the inbound traffic from J02M will be policed to 1Mbps on J58S.

- **Rewrite Marking**

Rewrite DSCP default marking will be configured at the egress on J58S. The goal is to mark HTTP traffic classified in the previous task with DSCP value of 101110. (To verify the configuration the DSCP default classifier will be applied on J11M).

- **Scheduler**

Scheduler maps will be configured on J58S to assign Best Effort traffic to 5% of the interface bandwidth and to guarantee Expedited Forwarding class 20% of the interface bandwidth.

Note: This section is not intended to follow “recommended practices” but to help clarify the concepts introduced in this chapter. Carefully plan any CoS implementation according to the requirements of each design.

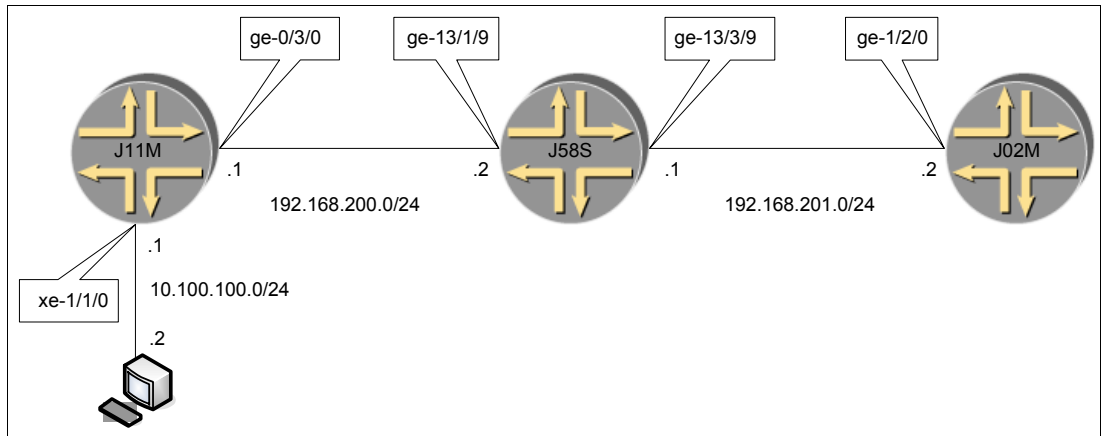


Figure 7-5 Case study diagram

Multifield classification and policing

In Example 7-31, we show firewall and policers settings on J11M.

Example 7-31 Multifield classification and policing configuration

```
{primary:node0}[edit]
ibm@J58S-1# set firewall filter MF-CLASSIFICATION term HTTP from protocol tcp
ibm@J58S-1# set firewall filter MF-CLASSIFICATION term HTTP from port 80
ibm@J58S-1# set firewall filter MF-CLASSIFICATION term HTTP then forwarding-class
expedited-forwarding
ibm@J58S-1# set firewall filter MF-CLASSIFICATION term OTHER then accept

ibm@J58S-1# set firewall policer LIMIT if-exceeding bandwidth-limit 1m
ibm@J58S-1# set firewall policer LIMIT if-exceeding burst-size-limit 15k
ibm@J58S-1# set firewall policer LIMIT then discard

ibm@J58S-1# show firewall filter MF-CLASSIFICATION
term HTTP {
  from {
    protocol tcp;
    port 80;
  }
  then forwarding-class expedited-forwarding;
}
term OTHER {
  then accept;
}

ibm@J58S-1# show firewall policer LIMIT
if-exceeding {
  bandwidth-limit 1m;
  burst-size-limit 15k;
}
then discard;

ibm@J58S-1# commit

{primary:node0}[edit]
```



```

ibm@J58S-1# set interfaces ge-13/3/9 unit 0 family inet filter input
MF-CLASSIFICATION
ibm@J58S-1# set interfaces ge-13/3/9 unit 0 family inet policer input LIMIT
ibm@J58S-1# commit

ibm@J58S-1# show interfaces ge-13/3/9
unit 0 {
    family inet {
        filter {
            input MF-CLASSIFICATION;
        }
        policer {
            input LIMIT;
        }
        address 192.168.201.1/24;
    }
}

```

To verify the configuration, check counters on the output interface of J58S, as we show in the following outputs.

1. First, clear counters on the interface:

```

{primary:node0}[edit]
ibm@J58S-1# run clear interfaces statistics ge-13/1/9
Verify queue counters:
{primary:node0}[edit]
ibm@J58S-1# run show interfaces ge-13/1/9 detail | find "Queue counters"
Queue counters:      Queued packets  Transmitted packets  Dropped packets
0 best-effort        1              1                    0
1 expedited-fo       0              0                    0
2 assured-forw       0              0                    0
3 network-cont       0              0                    0

```

2. Telnet from J02 to simulate TCP traffic on port 80:

```

[edit]
ibm@J02M# run telnet 192.168.200.1 port 80
Trying 192.168.200.1...
telnet: connect to address 192.168.200.1: Connection refused
telnet: Unable to connect to remote host

```

3. Verify queue counters:

```

{primary:node0}[edit]
ibm@J58S-1# run show interfaces ge-13/1/9 detail | find "Queue counters"
Queue counters:      Queued packets  Transmitted packets  Dropped packets
0 best-effort        1              1                    0
1 expedited-fo       1              1                    0
2 assured-forw       0              0                    0
3 network-cont       0              0                    0

```

As expected, you can see that one packet was placed in the Expedited Forwarding queue.

4. Ping from J02M to verify policer configuration:

```

[edit]
ibm@J02M# run ping 192.168.200.1 size 1500 rapid count 100
PING 192.168.200.1 (192.168.200.1): 1500 data bytes

```

```

!!!!!!.!!!!!!.!!!!!!.!!!!!!.!!!!!!.!!!!!!.!!!!!!.!!!!!!.!!!!!!.!!!!!!
!!.!!!!!!.!!!!!!.!!!!!!
--- 192.168.200.1 ping statistics ---
100 packets transmitted, 83 packets received, 17% packet loss
round-trip min/avg/max/stddev = 1.069/1.875/25.334/3.248 ms

```

5. Verify the policer on J58S:

```

{primary:node0}[edit]
ibm@J58S-1# run show policer LIMIT-ge-13/3/9.0-inet-i
Policers:
Name                                     Packets
LIMIT-ge-13/3/9.0-inet-i               41

```

Rewrite marking

HTTP was already assigned to Expedited Forwarding forwarding class on J58S. We must apply the default DSCP rewrite table to the egress interface on J58S as shown in Example 7-32.

Example 7-32 Assigning DSCP default rewrite rules

```

{primary:node0}[edit]
ibm@J58S-1# set class-of-service interfaces ge-13/1/9 unit 0 rewrite-rules dscp
default

{primary:node0}[edit]
ibm@J58S-1# commit

```

1. To verify the configuration, first check the default DSCP rewrite table:

```

{primary:node0}[edit]
ibm@J58S-1# run show class-of-service rewrite-rule type dscp
Rewrite rule: dscp-default, Code point type: dscp, Index: 31
  Forwarding class      Loss priority  Code point
  best-effort           low           000000
  best-effort           high          000000
  expedited-forwarding  low           101110
  expedited-forwarding  high          101110
  assured-forwarding    low           001010
  assured-forwarding    high          001100
  network-control       low           110000
  network-control       high          111000

```

2. Verify that the table was applied to the specified interface:

```

{primary:node0}[edit]
ibm@J58S-1# run show class-of-service interface ge-13/1/9
Physical interface: ge-13/1/9, Index: 194
Queues supported: 4, Queues in use: 4
  Scheduler map: <default>, Index: 2
  Chassis scheduler map: <default-chassis>, Index: 4

Logical interface: ge-13/1/9.0, Index: 81
Object      Name                Type      Index
Rewrite     dscp-default        dscp      31
Classifier  ipprec-compatibility ip         13

```

You can also generate traffic to the end host and check the interface counters at the egress port on J11M.

3. Verify counters on J11M egress port before generating traffic:

```
{master}[edit]
ibm@J11M-re0# run show interfaces xe-1/1/0 detail | find "Queue counters"
Queue counters:      Queued packets  Transmitted packets  Dropped packets
0 best-effort        5          5          0
1 expedited-fo       0          0          0
2 assured-forw       0          0          0
3 network-cont       1          1          0
```

4. Telnet from J02M to the endstation on port 80:

```
ibm@J02M> telnet 10.100.100.2 port 80
Trying 10.100.100.2...
```

5. Again, verify counters on J11M egress port:

```
{master}[edit]
ibm@J11M-re0# run show interfaces xe-1/1/0 detail | find "Queue counters"
Queue counters:      Queued packets  Transmitted packets  Dropped packets
0 best-effort        6          6          0
1 expedited-fo       0          0          0
2 assured-forw       0          0          0
3 network-cont       2          2          0
```

The Expedited Forwarding queue counters remained in 0 because the correct behavior aggregate (BA) classifier needs to be applied at the ingress interface on J11M.

6. Check what is the classifier currently applied to the ingress interface:

```
{master}[edit]
ibm@J11M-re0# run show class-of-service interface ge-0/3/0
Physical interface: ge-0/3/0, Index: 182
Queues supported: 8, Queues in use: 4
Scheduler map: <default>, Index: 2

Logical interface: ge-0/3/0.0, Index: 83
Object      Name      Type
Index
Classifier   ipprec-compatibility  ip
13
```

7. Apply the dscp default classifier in the ingress interface on J11M:

```
{master}[edit]
ibm@J11M-re0# set class-of-service interfaces ge-0/3/0 unit 0 classifiers dscp
default
```

```
{master}[edit]
ibm@J11M-re0# commit
```

```
{master}[edit]
ibm@J11M-re0# run show class-of-service interface ge-0/3/0
Physical interface: ge-0/3/0, Index: 182
Queues supported: 8, Queues in use: 4
Scheduler map: <default>, Index: 2
```

```
Logical interface: ge-0/3/0.0, Index: 83
```

Object	Name	Type
Index		
Classifier	dscp-default	dscp
7		

8. Telnet again from J02M to the endstation on port 80:

```
ibm@J02M> telnet 10.100.100.2 port 80
Trying 10.100.100.2...
```

9. Verify counters on J11M egress port:

```
{master}[edit]
ibm@J11M-re0# run show interfaces xe-1/1/0 detail | find "Queue counters"
Queue counters:      Queued packets  Transmitted packets  Dropped packets
0 best-effort        17                17                   0
1 expedited-fo       2                  2                    0
2 assured-forw       0                  0                    0
3 network-cont       30                30                   0
```

As expected, TCP port 80 packets were assigned to the Expedited Forwarding Queue.

Scheduler

In Example 7-33, we show how to configure the scheduler and scheduler-map.

Example 7-33 Scheduler configuration

```
{primary:node0}[edit]
ibm@J58S-1# set class-of-service schedulers BE transmit-rate percent 5
ibm@J58S-1# set class-of-service schedulers EF transmit-rate percent 20
ibm@J58S-1# set class-of-service scheduler-maps SCHED-MAP forwarding-class
best-effort scheduler BE
ibm@J58S-1# set class-of-service scheduler-maps SCHED-MAP forwarding-class
expedited-forwarding scheduler EF
ibm@J58S-1# set class-of-service interfaces ge-13/1/9 scheduler-map SCHED-MAP
ibm@J58S-1# commit
```

To set up the scheduler and scheduler-map:

1. Verify the configuration of the scheduler map:

```
{primary:node0}[edit]
ibm@J58S-1# run show class-of-service scheduler-map SCHED-MAP
Scheduler map: SCHED-MAP, Index: 25389
```

```
Scheduler: BE, Forwarding class: best-effort, Index: 2053
  Transmit rate: 5 percent, Rate Limit: none, Buffer size: remainder, Buffer
Limit: none, Priority: low
  Excess Priority: unspecified
  Drop profiles:
    Loss priority  Protocol  Index  Name
    Low           any       1      <default-drop-profile>
    Medium low    any       1      <default-drop-profile>
    Medium high   any       1      <default-drop-profile>
    High          any       1      <default-drop-profile>
```

```
Scheduler: EF, Forwarding class: expedited-forwarding, Index: 2278
```

```

Transmit rate: 20 percent, Rate Limit: none, Buffer size: remainder, Buffer
Limit: none, Priority: low
Excess Priority: unspecified
Drop profiles:
  Loss priority  Protocol  Index  Name
  Low           any       1      <default-drop-profile>
  Medium low    any       1      <default-drop-profile>
  Medium high   any       1      <default-drop-profile>
  High          any       1      <default-drop-profile>

```

2. Verify if it is correctly applied to the interface:

```

{primary:node0}[edit]
ibm@J58S-1# run show class-of-service interface ge-13/1/9
Physical interface: ge-13/1/9, Index: 194
Queues supported: 4, Queues in use: 4
Scheduler map: SCHED-MAP, Index: 25389
Chassis scheduler map: <default-chassis>, Index: 4

Logical interface: ge-13/1/9.0, Index: 81
  Object          Name          Type
Index
  Rewrite         dscp-default  dscp
31
  Classifier      ipprec-compatibility  ip
13

```

7.12 More information

For more information about Class of Service on IBM j-type s-series, refer to *JUNOS Software Interfaces and Routing Configuration Guide*:

<http://www.juniper.net/techpubs/software/junos-security/junos-security10.1/junos-security-swconfig-interfaces-and-routing/junos-security-swconfig-interfaces-and-routing.pdf>

For more information about Differentiated Services Model, refer to RFC 2475:

<http://www.ietf.org/rfc/rfc2475.txt>

For more information about Single Rate Three Color Marker, refer to RFC 2697:

<http://www.ietf.org/rfc/rfc2697.txt>

For more information about Two Rate Color Marker, refer to RFC 2698:

<http://www.ietf.org/rfc/rfc2698.txt>



Network security

In this chapter, we present an overview and configuration commands of network security capabilities for IBM j-type appliances.

Basic understanding of networking concepts is assumed. Refer to *IBM j-type Data Center Networking Introduction*, SG24-7820 and Chapter 1, “Fundamentals of Ethernet networking” on page 1 for a short introduction.

The topics that we discuss in this section are:

- ▶ Firewall filters (stateless)
- ▶ Security zones
- ▶ Security policies
- ▶ Network address translation
- ▶ Virtual Private Networks
- ▶ Authentication options

In this chapter, we present usual configuration scenarios for data center implementations, and as such not all options are described. For complete and advanced information read the *JUNOS Software Security Configuration Guide, Release 10.1*, which is available at

<http://www.juniper.net/techpubs/software/junos-security/junos-security10.1/junos-security-swconfig-security/junos-security-swconfig-security.pdf>

8.1 Sample topology

Unless stated otherwise, we use the topology presented in Figure 8-1 for all of our examples from this chapter.

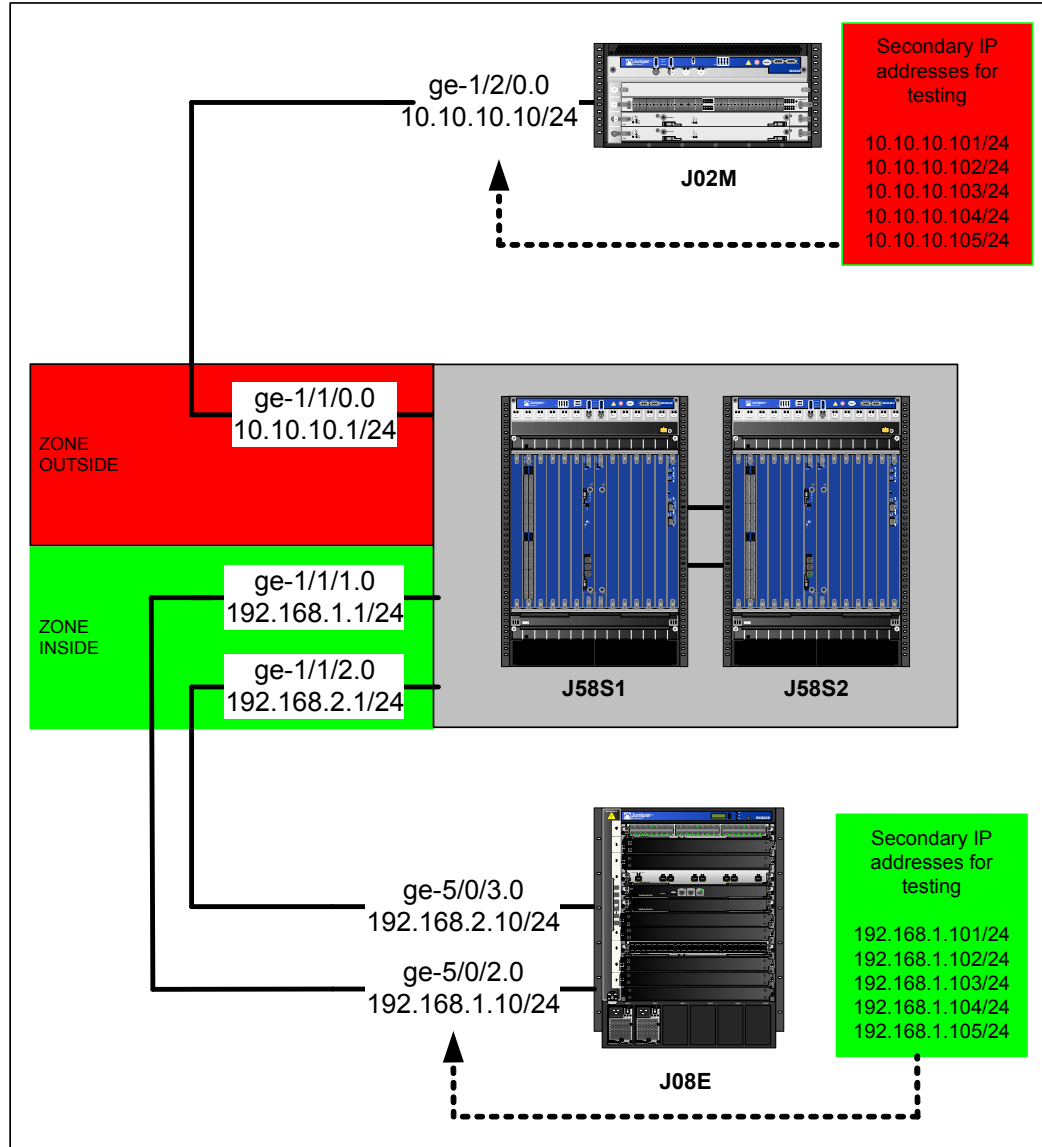


Figure 8-1 Topology used for examples in this chapter

We use a pair of IBM J58S security appliances configured in a redundant “chassis cluster” configuration. We connected an IBM J08E switch for simulating an inside network and hosts/servers and an IBM J02M router for simulation and outside network and servers.

8.2 Firewall filters (stateless)

This section presents a brief overview of stateless firewall filters and how to configure and monitor them on IBM j-type security appliances.

Firewall filters overview

A *stateless* firewall filter evaluates the contents of packets transiting the device from a source to a destination, or packets originating from, or destined for, the Routing Engine. Stateless firewall filters applied to the Routing Engine interface protect the processes and resources owned by the Routing Engine. A stateless firewall filter evaluates every packet, including fragmented packets.

A stateless firewall filter, often called a firewall filter or access control list (ACL), statically evaluates packet contents. In contrast, a stateful firewall filter uses connection state information derived from past communications and other applications to make dynamic control decisions.

A stateless firewall filter can filter packets transiting the device from a source to a destination, or packets originating from, or destined for, the Routing Engine. Stateless firewall filters applied to the Routing Engine interface protect the processes and resources owned by the Routing Engine.

You can apply a stateless firewall filter to an input or output interface, or to both. Every packet, including fragmented packets, are evaluated against stateless firewall filters.

Stateless firewall filter terms

All stateless firewall filters contain one or more terms, and each term consists of two components—match conditions and actions. The match conditions define the values or fields that the packet must contain to be considered a match. If a packet is a match, the corresponding action is taken. By default, a packet that does not match a firewall filter is discarded.

Chained stateless firewall filters

You can configure a stateless firewall filter within the term of another filter. This method enables you to add common terms to multiple filters without having to modify all filter definitions. You can configure one filter with the desired common terms, and configure this filter as a term in other filters. Consequently, to make a change in these common terms, you need to modify only one filter that contains the common terms, instead of multiple filters.

Planning a stateless firewall filter

Before creating a stateless firewall filter and applying it to an interface, determine what you want the firewall filter to accomplish and how to use its match conditions and actions to achieve your goal. Also, make sure you understand how packets are matched and the default action of the resulting firewall filter.

To configure a stateless firewall filter, determine the following:

- ▶ Purpose of the firewall filter—for example, to limit traffic to certain protocols, IP source or destination addresses, or data rates, or to prevent denial-of-service (DoS) attacks.
- ▶ Appropriate match conditions. The packet header fields to match—for example, IP header fields (such as source and destination IP addresses, protocols, and IP options), TCP header fields (such as source and destination ports and flags), and ICMP header fields (such as ICMP packet type and code).
- ▶ Action to take if a match occurs—for example, accept, discard, or evaluate the next term.
- ▶ (Optional) Action modifiers. Additional actions to take if a packet matches—for example, count, log, rate limit, or police a packet.
- ▶ Interface on which the firewall filter is applied. The input or output side, or both sides, of the Routing Engine interface or a non-Routing Engine interface.

Stateless firewall filter strategies

For best results, use the following sections to plan the purpose and contents of a stateless firewall filter before starting configuration.

Strategy for a typical stateless firewall filter

A primary goal of a typical stateless firewall filter is to protect the Routing Engine processes and resources from malicious or entrusted packets. You can configure a firewall filter to restrict traffic destined for the Routing Engine based on its source, protocol, and application. In addition, you can limit the traffic rate of packets destined for the Routing Engine to protect against flood, or denial-of-service, attacks.

Note: Stateless firewall filters are also useful for dropping traffic without having the flow engine have to inspect it. You can use Screens to block fragments optionally. Simple Filters can be used to do stateless firewall filters in hardware.

Handling packet fragments

You can configure a stateless firewall filter to address special circumstances associated with fragmented packets destined for the Routing Engine. Because the device evaluates every packet against a firewall filter (including fragments), you must configure the filter to accommodate fragments that do not contain packet header information. Otherwise, the filter discards all but the first fragment of a fragmented packet.

Note: We can also use firewall filters for filtering transit traffic, but in case of IBM j-type security appliances, stateful firewall is available and recommended for transit traffic.

To define the match condition for a firewall filter, use the command:

```
{primary:node0}[edit]
ibm@J58S-1# set firewall family inet filter filter_name term term_name from
match_options
```

In the command:

- ▶ Filter_name is the name of the firewall filter
- ▶ Term_name is the name of the specific term from the firewall filter
- ▶ Match_options are the attributes of the IP packets for which we apply the filter

The common match options are **source-address**, **destination-address**, **source-port**, **destination-port**, and **protocol**.

To define an action for the defined match, use the command:

```
{primary:node0}[edit]
ibm@J58S-1# set firewall family inet filter filter_name term term_name then action
```

In the command:

- ▶ Filter_name is the name of the firewall filter
- ▶ Term_name is the name of the specific term from the firewall filter
- ▶ Action is the action to take when a packet matches the condition for this term

To apply a filter for protecting the Routing Engine, use the command:

```
{primary:node0}[edit]
ibm@J58S-1# set interfaces lo0.0 family inet filter input filter_name
```

In the command, filter_name is the name of the filter that we want to apply.

To verify the firewall configuration, use the command:

```
{primary:node0}[edit]
ibm@J58S-1# show firewall family inet
```

To verify the operation of firewall filter, use the command:

```
{primary:node0}
ibm@J58S-1> show firewall
```

To view the logs of the firewall filter, use the command:

```
{primary:node0}
ibm@J58S-1> show firewall log
```

Firewall filter configuration example

In Example 8-1, we present a firewall filter configuration for protecting the router engine with the following characteristics:

- ▶ Permit and log ssh and telnet access only from 10.1.1.0/24 range
- ▶ Permit udp traffic from 207.17.137.28
- ▶ Permit icmp echo and icmp echo response traffic from all addresses
- ▶ Permit ntp traffic from 149.20.68.16
- ▶ Permit ospf protocol
- ▶ Permit snmp and snmp-trap traffic from 10.1.1.0/24
- ▶ Reject with tcp-reset and count all other traffic in **denied** counter

Example 8-1 Firewall filter configuration

```
{primary:node0}[edit]
ibm@J58S-1# edit firewall family inet filter protect-re

{primary:node0}[edit firewall family inet filter protect-re]
ibm@J58S-1#

{primary:node0}[edit firewall family inet filter protect-re]
ibm@J58S-1# set term permit-ssh from source-address 10.1.1.0/24

{primary:node0}[edit firewall family inet filter protect-re]
ibm@J58S-1# set term permit-ssh from protocol tcp

{primary:node0}[edit firewall family inet filter protect-re]
ibm@J58S-1# set term permit-ssh from destination-port ssh

{primary:node0}[edit firewall family inet filter protect-re]
ibm@J58S-1# set term permit-ssh from destination-port telnet

{primary:node0}[edit firewall family inet filter protect-re]
ibm@J58S-1# set term permit-ssh then accept

{primary:node0}[edit firewall family inet filter protect-re]
ibm@J58S-1# set term permit-udp from source-address 207.17.137.28

{primary:node0}[edit firewall family inet filter protect-re]
ibm@J58S-1# set term permit-udp from protocol udp

{primary:node0}[edit firewall family inet filter protect-re]
ibm@J58S-1# set term permit-udp then accept
```

```

{primary:node0}[edit firewall family inet filter protect-re]
ibm@J58S-1# set term permit-icmp from protocol icmp

{primary:node0}[edit firewall family inet filter protect-re]
ibm@J58S-1# set term permit-icmp from icmp-type echo-reply

{primary:node0}[edit firewall family inet filter protect-re]
ibm@J58S-1# set term permit-icmp from icmp-type echo-request

{primary:node0}[edit firewall family inet filter protect-re]
ibm@J58S-1# set term permit-icmp then accept

{primary:node0}[edit firewall family inet filter protect-re]
ibm@J58S-1# set term permit-ntp from source-address 149.20.68.16

{primary:node0}[edit firewall family inet filter protect-re]
ibm@J58S-1# set term permit-ntp from protocol udp

{primary:node0}[edit firewall family inet filter protect-re]
ibm@J58S-1# set term permit-ntp from port ntp

{primary:node0}[edit firewall family inet filter protect-re]
ibm@J58S-1# set term permit-ntp then accept

{primary:node0}[edit firewall family inet filter protect-re]
ibm@J58S-1# set term permit-ospf from protocol ospf

{primary:node0}[edit firewall family inet filter protect-re]
ibm@J58S-1# set term permit-ospf then accept

{primary:node0}[edit firewall family inet filter protect-re]
ibm@J58S-1# set term permit-snmp from protocol udp

{primary:node0}[edit firewall family inet filter protect-re]
ibm@J58S-1# set term permit-snmp from port snmp

{primary:node0}[edit firewall family inet filter protect-re]
ibm@J58S-1# set term permit-snmp from port snmptrap

{primary:node0}[edit firewall family inet filter protect-re]
ibm@J58S-1# set term permit-snmp then accept

{primary:node0}[edit firewall family inet filter protect-re]
ibm@J58S-1# set term deny-and-count from source-address 0.0.0.0/0

{primary:node0}[edit firewall family inet filter protect-re]
ibm@J58S-1# set term deny-and-count then count denied

{primary:node0}[edit firewall family inet filter protect-re]
ibm@J58S-1# set term deny-and-count then reject tcp-reset

{primary:node0}[edit firewall family inet filter protect-re]
ibm@J58S-1# top set interfaces lo0.0 family inet filter input protect-re

{primary:node0}[edit firewall family inet filter protect-re]

```

```

ibm@J58S-1# show
term permit-ssh {
    from {
        source-address {
            10.1.1.0/24;
        }
        protocol tcp;
        destination-port [ ssh telnet ];
    }
    then accept;
}
term permit-udp {
    from {
        source-address {
            207.17.137.28/32;
        }
        protocol udp;
    }
    then accept;
}
term permit-icmp {
    from {
        protocol icmp;
        icmp-type [ echo-reply echo-request ];
    }
    then accept;
}
term permit-ntp {
    from {
        source-address {
            149.20.68.16/32;
        }
        protocol udp;
        port ntp;
    }
    then accept;
}
term permit-ospf {
    from {
        protocol ospf;
    }
    then accept;
}
term permit-snmp {
    from {
        protocol udp;
        port [ snmp snmptrap ];
    }
    then accept;
}
term deny-and-count {
    from {
        source-address {
            0.0.0.0/0;
        }
    }
}

```

```

    }
    then {
        count denied;
        reject tcp-reset;
    }
}

```

```

{primary:node0}[edit firewall family inet filter protect-re]
ibm@J58S-1# top

```

```

{primary:node0}[edit]
ibm@J58S-1# commit
node0:
commit complete
node1:
commit complete

```

```

{primary:node0}[edit]
ibm@J58S-1# run show firewall filter protect-re

```

Filter: protect-re

Counters:

Name	Bytes	Packets
denied	14151	146

```

{primary:node0}[edit]
ibm@J58S-1# run show interfaces lo0.0 detail
Logical interface lo0.0 (Index 64) (SNMP ifIndex 16) (Generation 129)
Flags: SNMP-Traps Encapsulation: Unspecified
Traffic statistics:
  Input bytes :          10611
  Output bytes :          10611
  Input packets:           161
  Output packets:          161
Local statistics:
  Input bytes :          10611
  Output bytes :          10611
  Input packets:           161
  Output packets:          161
Security: Zone: Null
Flow Statistics :
Flow Input statistics :
  Self packets :          0
  ICMP packets :          0
  VPN packets :           0
  Multicast packets :      0
  Bytes permitted by policy : 0
  Connections established : 0
Flow Output statistics:
  Multicast packets :      0
  Bytes permitted by policy : 0
Flow error statistics (Packets dropped due to):
  Address spoofing:        0
  Authentication failed:   0
  Incoming NAT errors:     0

```

```

Invalid zone received packet:      0
Multiple user authentications:    0
Multiple incoming NAT:           0
No parent for a gate:            0
No one interested in self packets: 0
No minor session:                0
No more sessions:                0
No NAT gate:                     0
No route present:                0
No SA for incoming SPI:          0
No tunnel found:                 0
No session for a gate:           0
No zone or NULL zone binding     0
Policy denied:                   0
Security association not active:  0
TCP sequence number out of window: 0
Syn-attack protection:           0
User authentication errors:       0
Protocol inet, MTU: Unlimited, Generation: 136, Route table: 0
  Input Filters: protect-re, <----- firewall filter applied on interface
  Addresses, Flags: Is-Default Is-Primary
    Destination: Unspecified, Local: 10.8.8.2, Broadcast: Unspecified,
Generation: 183

{primary:node0}[edit]
ibm@J58S-1#

```

8.3 Security zones

This section presents a brief overview of security zones and how to configure and monitor them on IBM j-type security appliances.

8.3.1 Security zones and interfaces overview

Interfaces act as a doorway through which traffic enters and exits an IBM j-type security appliance. Many interfaces can share exactly the same security requirements; however, different interfaces can also have different security requirements for inbound and outbound data packets. Interfaces with identical security requirements can be grouped together into a single security zone.

A security zone is a collection of one or more network segments requiring the regulation of inbound and outbound traffic through policies.

Security zones are logical entities to which one or more interfaces are bound. With many types of IBM j-type security appliances, you can define multiple security zones, the exact number of which you determine based on your network needs.

On a single device, you can configure multiple security zones, dividing the network into segments to which you can apply various security options to satisfy the needs of each segment. At a minimum, you must define two security zones, basically to protect one area of the network from the other. On some security platforms, you can define many security zones,

bringing finer granularity to your network security design and without deploying multiple security appliances to do so.

From the perspective of security policies, traffic enters into one security zone and goes out on another security zone. This combination of a from-zone and a to-zone is defined as a context. Each context contains an ordered list of policies.

Understanding security zone interfaces

An interface for a security zone can be thought of as a doorway through which TCP/IP traffic can pass between that zone and any other zone.

Through the policies you define, you can permit traffic between zones to flow in one direction or in both. With the routes that you define, you specify the interfaces that traffic from one zone to another must use. Because you can bind multiple interfaces to a zone, the routes you chart are important for directing traffic to the interfaces of your choice.

Understanding interface ports

On IBM j-type security appliance interface ports are located on Physical Interface Cards (PIC) that are inserted in Flexible PIC concentrators.

IBM j-type security appliances use a naming convention for defining the interfaces that is similar to that of other platforms running under Junos software. Each physical port can have many logical interfaces configured with properties different from the port's other logical units.

Interfaces used for data traffic in Junos software are specified as type-fpc/pic/port.logical-unit-number. Table 8-1 presents the meaning of each option.

Table 8-1 Interface naming convention

Code	Description
type	<ul style="list-style-type: none">▶ ge: Gigabit Ethernet interface▶ xe: 10 Gigabit Ethernet interface▶ fe: Fast Ethernet interface
fpc	Flexible PIC Concentrator: <ul style="list-style-type: none">▶ FPC number indicates the slot number of the line card that contains the physical interface
pic	Physical Interface Card: <ul style="list-style-type: none">▶ PIC number represents its number on the PIC concentrator
port	Port number
logical-unit-number	Logical Unit Number of the physical interface

8.3.2 Security zones

In IBM j-type security appliances, we have two types of zones: functional zones and security zones.

Functional zones

A functional zone is used for special purposes, such as management interfaces. Currently, only the management (MGT) zone is supported. Management zones have the following properties:

- ▶ Management zones host management interfaces.

- ▶ Traffic entering management zones does not match policies; therefore, traffic cannot transit out of any other interface if it was received in the management interface.
- ▶ Management zones can only be used for dedicated management interfaces.

Security zones

Security zones are the building blocks for policies; they are logical entities to which one or more interfaces are bound. Security zones provide a means of distinguishing groups of hosts (user systems and other hosts, such as servers) and their resources from one another to apply different security measures to them.

Security zones have the following properties:

- ▶ **Policies:** Active security policies that enforce rules for the transit traffic, in terms of what traffic can pass through the firewall, and the actions that need to take place on the traffic as it passes through the firewall.
- ▶ **Screens:** A stateful firewall secures a network by inspecting, and then allowing or denying, all connection attempts that require passage from one security zone to another. For every security zone, and the management zone, you can enable a set of predefined screen options that detect and block various kinds of traffic that the device determines as potentially harmful.
- ▶ **Address books:** IP addresses and address sets that make up an address book to identify its members so that you can apply policies to them.
- ▶ **TCP-RST:** When this feature is enabled, the system sends a TCP segment with the RESET flag set when traffic arrives that does not match an existing session and does not have the SYNchronize flag set.
- ▶ **Interfaces:** List of interfaces in the zone.

Security zones have the following preconfigured zones:

- ▶ *Junos-global zone:* Defined in the Junos defaults and cannot be configured by the user. The global zone serves as a storage area for static NAT addresses and can be used in policies like any other security zone.
- ▶ *Trust zone:* Available only in the factory configuration and is used for initial connection to the device. After you commit a configuration, the trust zone can be overridden.

To configure an interface and its IP address use the command:

```
{primary:node0}[edit]
ibm@J58S-1# set interfaces interface family inet address IP_address/mask
```

In this command:

- ▶ Interface is the name of the interface in *type-fpc/pic/port.logical-unit-number* notation
- ▶ IP_address/mask is the assigned IP address and mask of the interface.

To configure a zone and assign an interface to it, use the command:

```
{primary:node0}[edit]
ibm@J58S-1# set security zones security-zone zone interfaces interface
```

In this command:

- ▶ Zone is the name of the created zone
- ▶ Interface is the name of the interface assigned to the zone specified in *type-fpc/pic/port.logical-unit-number* notation.

To verify the configuration of the zones, use the command:

```
{primary:node0}[edit]  
ibm@J58S-1# show security zones
```

To verify the status of zones, use the operational command:

```
{primary:node0}  
ibm@J58S-1> show security zones
```

To verify the status of the interface with regard to security zones, use the command:

```
{primary:node0}  
ibm@J58S-1> show interfaces [interface]
```

8.3.3 Host inbound traffic

We now discuss how to control Inbound Traffic Based on Traffic Types. This topic describes how to configure zones to specify the kinds of traffic that can reach the device from systems that are directly connected to its interfaces.

Note that:

- ▶ You can configure these parameters at the zone level, in which case they affect all interfaces of the zone, or at the interface level. (Interface configuration overrides that of the zone.)
- ▶ You must enable all expected host-inbound traffic. Inbound traffic from devices directly connected to the device's interfaces is dropped by default.
- ▶ You can also configure a zone's interfaces to allow for use by dynamic routing protocols.

This feature allows you to protect the device against attacks launched from systems that are directly connected to any of its interfaces. It also enables you to selectively configure the device so that administrators can manage it using certain applications on certain interfaces. You can prohibit use of other applications on the same or different interfaces of a zone. For example, most likely you want to ensure that outsiders do not use the Telnet application from the Internet to log into the device because you do not want them connecting to your system.

The supported system services are **http**, **rpm**, **traceroute**, **bootp**, **https**, **rsh**, **xnm-clear-text**, **dhcp**, **ike**, **snmp**, **xnm-ssl**, **finger**, **netconf**, **snmp-trap**, **ftp**, **ping**, **ssh**, **ident-reset**, **rlogin**, and **telnet**.

To permit host inbound services traffic in a zone, use the command:

```
{primary:node0}[edit]  
ibm@J58S-1# set security zones security-zone zone host-inbound-traffic  
system-services service
```

In the command:

- ▶ Zone is the name of the zone
- ▶ Service is the service to be allowed

To permit host inbound services traffic in a specific interface from a zone, use the command:

```
{primary:node0}[edit]  
ibm@J58S-1# set security zones security-zone zone interfaces interface  
host-inbound-traffic system-services service
```

In the command:

- ▶ Zone is the name of the zone
- ▶ Interface is the name of the interface assigned to the zone specified in type-fpc/pic/port.logical-unit-number notation
- ▶ Service is the service to be allowed

8.3.4 Control inbound traffic based on protocols

This topic describes the inbound system protocols on the specified zone or interface.

Any host-inbound traffic that corresponds to a protocol listed under the host-inbound traffic option is allowed. For example, if anywhere in the configuration, you map a protocol to a port number other than the default, you can specify the protocol on the host-inbound traffic option, and the new port number will be used.

The supported protocols are **igmp**, **pgm**, **sap**, **bfd**, **ldp**, **pim**, **vrrp**, **bgp**, **msdp**, **rip**, **nhrp**, **router-discovery**, **dvmrp**, **ospf**, and **rsvp**.

To permit host inbound protocol traffic in a zone, use the command:

```
{primary:node0}[edit]
ibm@J58S-1# set security zones security-zone zone host-inbound-traffic protocols
protocol
```

In the command:

- ▶ Zone is the name of the zone
- ▶ Protocol is the network protocol to be allowed

To permit host inbound protocol traffic in a specific interface from a zone, use the command:

```
{primary:node0}[edit]
ibm@J58S-1# set security zones security-zone zone interfaces interface
host-inbound-traffic protocols protocol
```

In the command:

- ▶ Zone is the name of the zone
- ▶ Interface is the name of the interface assigned to the zone specified in type-fpc/pic/port.logical-unit-number notation
- ▶ Protocol is the network protocol to be allowed

8.3.5 TCP-Reset parameters

When the TCP-RST feature is enabled, the system sends a TCP segment with the RESET flag set when traffic arrives that does not match an existing session and does not have the SYNchronize flag set.

To enable TCP-RST feature in a zone, use the command:

```
{primary:node0}[edit]
ibm@J58S-1# set security zones security-zone zone tcp-rst
```

In the command, zone is the name of the zone.

8.3.6 Address books and address sets

Each security zone contains an address book. Before you can set up policies between two zones, you must define the addresses for each of the zone's address books. To manage an address book with large numbers of addresses, you can create groups of addresses called address sets.

A security zone is a logical group of interfaces with identical security requirements. Each security zone contains an address book. Before you can set up policies between two zones, you must define the addresses for each of the zone's address books. A zone's address book must contain entries for the addressable networks and end hosts (and, thus, users) belonging to the zone.

The following guidelines apply to address books:

- ▶ An address book for a security zone contains the IP address or domain names of hosts and subnets whose traffic is either allowed, blocked, encrypted, or user-authenticated.
- ▶ Address books can have address sets. Each address set has a name and a list of address names.
- ▶ Addresses and address sets in the same zone must have distinct names.
- ▶ Addresses must conform to the security requirements of the zone.
- ▶ IP addresses can be configured as IPv4 addresses with the number of prefix bits, or as Domain Name System (DNS) names.
- ▶ The predefined address any is automatically created for each security zone.
- ▶ The address book of a security zone must contain all IP addresses that are reachable within that zone.

Policies contain both source and destination zones and addresses. An address is referred to in a policy by the name you give it in its zone's address book.

- ▶ When traffic is sent to a zone, the zone and address to which the traffic is sent are used as the destination zone and address-matching criteria in policies.
- ▶ When traffic is sent from a zone, the zone and address from which it is sent are used as the matching source zone and address in policies.

An address book can grow to contain large numbers of addresses and become difficult to manage. To manage an address book with large numbers of addresses, you can create groups of addresses called address sets. You can reference an address set in a policy as you do an individual address book entry.

To configure address book entries for a zone, use the following commands:

```
{primary:node0}[edit]
ibm@J58S-1# set security zones security-zone zone address-book address
address-name IP_address/prefix
```

In the command:

- ▶ Zone is the name of the zone
- ▶ Address-name is the name of the address book
- ▶ IP_address/prefix is the IP address and prefix that you want to have in this address book name

To configure address-set entries for a zone, use the following commands:

```
{primary:node0}[edit]
ibm@J58S-1# set security zones security-zone zone address-book address-set
address-set-name address address-name
```

In the command:

- ▶ Zone is the name of the zone
- ▶ Address-set-name is the name of the address-set
- ▶ Address-name is the name of the address book that you want to have in this address-set name

8.3.7 Zone configuration example

Example 8-2 presents configuration for two security zones with the following characteristics:

- ▶ Zone outside:
 - This zone has interface ge-1/1/0.0
 - IP address of interface ge-1/1/0.0 is 10.10.10.1/24
 - Tcp reset feature enabled
 - Inbound permitted services are: **ike** and **ping**
 - Address-book **out-smtp** with IP address of 10.10.10.100
 - Address-book **out-pop3** with IP address of 10.10.10.101
 - Address-book **out-imap** with IP address of 10.10.10.102
 - Address-set **out-mail** with address-books **out-smtp**, **out-pop3**, and **out-imap**
- ▶ Zone inside:
 - This zone has interface ge-1/1/1.0 and ge-1/1/2.0
 - IP address of interface ge-1/1/1.0 is 192.168.1.1/24
 - IP address of interface ge-1/1/2.0 is 192.168.2.1/24
 - Inbound permitted services are: **ssh**, **https**, **ping**, **snmp**, **traceroute**, and **netconf**
 - Inbound permitted protocols are: **ospf** and **bgp**
 - Address-book **local-lan** with IP subnet of 192.168.0.0/16

Example 8-2 Security zone configuration

```
{primary:node0}[edit]
ibm@J58S-1# set interfaces ge-1/1/0.0 family inet address 10.10.10.1/24

{primary:node0}[edit]
ibm@J58S-1# set interfaces ge-1/1/1.0 family inet address 192.168.1.1/24

{primary:node0}[edit]
ibm@J58S-1# set interfaces ge-1/1/2.0 family inet address 192.168.2.1/24

{primary:node0}[edit]
ibm@J58S-1# set security zones security-zone outside interfaces ge-1/1/0.0

{primary:node0}[edit]
ibm@J58S-1# set security zones security-zone outside tcp-rst

{primary:node0}[edit]
ibm@J58S-1# set security zones security-zone outside host-inbound-traffic
system-services ping
```

```

{primary:node0}[edit]
ibm@J58S-1# set security zones security-zone outside host-inbound-traffic
system-services ike

{primary:node0}[edit]
ibm@J58S-1# set security zones security-zone outside address-book address out-smtp
10.10.10.100/32

{primary:node0}[edit]
ibm@J58S-1# set security zones security-zone outside address-book address out-pop3
10.10.10.101/32

{primary:node0}[edit]
ibm@J58S-1# set security zones security-zone outside address-book address out-imap
10.10.10.102/32

{primary:node0}[edit]
ibm@J58S-1# set security zones security-zone outside address-book address-set
out-mail address out-smtp

{primary:node0}[edit]
ibm@J58S-1# set security zones security-zone outside address-book address-set
out-mail address out-pop3

{primary:node0}[edit]
ibm@J58S-1# set security zones security-zone outside address-book address-set
out-mail address out-imap

{primary:node0}[edit]
ibm@J58S-1# set security zones security-zone inside interfaces ge-1/1/1.0

{primary:node0}[edit]
ibm@J58S-1# set security zones security-zone inside interfaces ge-1/1/2.0

{primary:node0}[edit]
ibm@J58S-1# edit security zones security-zone inside host-inbound-traffic

{primary:node0}[edit security zones security-zone inside host-inbound-traffic]
ibm@J58S-1# set system-services ssh

{primary:node0}[edit security zones security-zone inside host-inbound-traffic]
ibm@J58S-1# set system-services https

{primary:node0}[edit security zones security-zone inside host-inbound-traffic]
ibm@J58S-1# set system-services ping

{primary:node0}[edit security zones security-zone inside host-inbound-traffic]
ibm@J58S-1# set system-services snmp

{primary:node0}[edit security zones security-zone inside host-inbound-traffic]
ibm@J58S-1# set system-services traceroute

{primary:node0}[edit security zones security-zone inside host-inbound-traffic]
ibm@J58S-1# set system-services netconf

```

```
{primary:node0}[edit security zones security-zone inside host-inbound-traffic]
ibm@J58S-1# set protocols ospf
```

```
{primary:node0}[edit security zones security-zone inside host-inbound-traffic]
ibm@J58S-1# set protocols bgp
```

```
{primary:node0}[edit security zones security-zone inside host-inbound-traffic]
ibm@J58S-1# up
```

```
{primary:node0}[edit security zones security-zone inside]
ibm@J58S-1# set address-book address local-lan 192.168.0.0/16
```

```
{primary:node0}[edit security zones security-zone inside]
ibm@J58S-1# top
```

```
{primary:node0}[edit]
ibm@J58S-1# show security zones
security-zone outside {
    tcp-rst;
    address-book {
        address out-smtp 10.10.10.100/32;
        address out-pop3 10.10.10.101/32;
        address out-imap 10.10.10.102/32;
        address-set out-mail {
            address out-smtp;
            address out-pop3;
            address out-imap;
        }
    }
    host-inbound-traffic {
        system-services {
            ping;
            ike;
        }
    }
    interfaces {
        ge-1/1/0.0;
    }
}
security-zone inside {
    address-book {
        address local-lan 192.168.0.0/16;
    }
    host-inbound-traffic {
        system-services {
            ssh;
            https;
            ping;
            snmp;
            traceroute;
            netconf;
        }
        protocols {
            ospf;
            bgp;
        }
    }
}
```

```

    }
  }
  interfaces {
    ge-1/1/1.0;
    ge-1/1/2.0;
  }
}

{primary:node0}[edit]
ibm@J58S-1# commit
node0:
configuration check succeeds
node1:
commit complete
node0:
commit complete

{primary:node0}[edit]
ibm@J58S-1#
ibm@J58S-1# run show security zones detail
node0:
-----

Security zone: inside
  Send reset for non-SYN session TCP packets: Off
  Policy configurable: Yes
  Interfaces bound: 2
  Interfaces:
    ge-1/1/1.0
    ge-1/1/2.0

Security zone: junos-global
  Send reset for non-SYN session TCP packets: Off
  Policy configurable: Yes
  Interfaces bound: 0
  Interfaces:

Security zone: outside
  Send reset for non-SYN session TCP packets: On
  Policy configurable: Yes
  Interfaces bound: 1
  Interfaces:
    ge-1/1/0.0

{primary:node0}[edit]
ibm@J58S-1# run show interfaces ge-1/1/0.0
Logical interface ge-1/1/0.0 (Index 82) (SNMP ifIndex 605)
  Flags: SNMP-Traps 0x4000000 Encapsulation: ENET2
  Input packets : 0
  Output packets: 0
  Security: Zone: outside <-----security zone assignment
  Allowed host-inbound traffic : ike ping <----- inbound allowed traffic
  Protocol inet, MTU: 1500
  Addresses, Flags: Is-Preferred Is-Primary

```


Destination: 10.10.10/24, Local: 10.10.10.1, Broadcast: 10.10.10.255
Protocol multiservice, MTU: Unlimited

{primary:node0}[edit]
ibm@J58S-1#

8.4 Security policies

This section presents a brief overview of security policies and how to configure and monitor them on IBM j-type security appliances.

8.4.1 Security policies overview

With the advent of the Internet, the need for a secure network has become vital for businesses with an Internet connection. Before a network can be secured for a business, a network security policy has to outline all the network resources within that business and identify the required security level for those resources. The network security policy also defines the security threats and the actions taken for such threats. Junos software stateful firewall policy provides a set of tools to network administrators, enabling them to implement network security for their organizations.

In a Junos software stateful firewall, the security policies enforce rules for the transit traffic, in terms of what traffic can pass through the firewall, and the actions that need to take place on the traffic as it passes through the firewall. From the perspective of security policies, the traffic enters one security zone and exits another security zone. This combination of a *from-zone* and *to-zone* is called a *context*. Each context contains an *ordered list of policies*.

A security policy, which can be configured from the user interface, controls the traffic flow from one zone to another zone by defining the kind(s) of traffic permitted from specified IP sources to specified IP destinations at scheduled times.

Policies allow you to deny, permit, reject (deny and send a TCP RST or ICMP port unreachable message to the source host), encrypt and decrypt, authenticate, prioritize, schedule, filter, and monitor the traffic attempting to cross from one security zone to another. You decide which users and what data can enter and exit, and when and where they can go.

IBM j-type security appliances secures a network by inspecting, and then allowing or denying, all connection attempts that require passage from one security zone to another.

By default, a device denies all traffic in all directions. Through the creation of policies, you can control the traffic flow from zone to zone by defining the kinds of traffic permitted to pass from specified sources to specified destinations at scheduled times.

At the broadest level, you can allow all kinds of traffic from any source in one zone to any destination in all other zones without any scheduling restrictions. At the narrowest level, you can create a policy that allows only one kind of traffic between a specified host in one zone and another specified host in another zone during a scheduled interval of time.

Understanding security policy rules

The security policy applies the security rules to the transit traffic within a context (from-zone to to-zone). Each policy is uniquely identified by its name. The traffic is classified by matching its source and destination zones, the source and destination addresses, and the application that the traffic carries in its protocol headers with the policy database in the data plane.

Each policy is associated with the following characteristics:

- ▶ A source zone
- ▶ A destination zone
- ▶ One or many source address names or address set names
- ▶ One or many destination address names or address set names
- ▶ One or many application names or application set names

These characteristics are called the match criteria. Each policy also has actions associated with it: permit, deny, and reject. You must specify the match condition arguments when you configure a policy, source address, destination address, and application name. If you do not want to specify a specific application, enter any as the default application, indicating all possible applications.

For example, if you do not supply an application name, the policy is installed with the application as a wildcard (default). Therefore, any data traffic that matches the rest of the parameters in a given policy matches the policy regardless of the application type of the data traffic.

The action of the first policy that the traffic matches is applied to the packet. If there is no matching policy, the packet is dropped. Policies are searched from top to bottom, so it is a good idea to place more specific policies near the top of the list. You must also place IPsec VPN tunnel policies near the top. Place the more general policies, such as one that allows certain users access to all Internet applications, at the bottom of the list.

Policies are applied after the packet has passed through the firewall's screens and the system has looked up its route. The packet's destination address determines its destination zone.

When you are creating a policy, the following policy rules apply:

- ▶ Security policies are configured in a from-zone to a to-zone direction. Under a specific zone direction, each security policy contains a name, match criteria, an action, and miscellaneous options.
- ▶ The policy name, match criteria, and action are required.
- ▶ The policy name is a keyword.
- ▶ The source address in the match criteria is composed of one or more address names or address set names in the from-zone.
- ▶ The destination address of the match criteria is composed of one or more address names or address set names in the to-zone.
- ▶ The application name in the match criteria is composed of the name of one or more applications or application sets.
- ▶ One of the following actions is required: permit, deny, or reject.
- ▶ When logging is enabled, the system logs at session close time by default. You can enable logging at session creation, too.
- ▶ When the count alarm is turned on, you can, optionally, specify alarm thresholds in bytes per second and kilobytes per minute.
- ▶ Each policy optionally indicates whether it allows NAT translation, does not allow NAT translation, or does not care.
- ▶ Address names cannot begin with the following reserved prefixes. These are used only for address NAT configuration:
 - Static_nat_
 - Incoming_nat_

- Junos_
- Application names cannot begin with the *junos_* reserved prefix.

Understanding security policy elements

A policy permits, denies, or tunnels specified types of traffic unidirectionally between two points.

To define a policy, you need:

- ▶ An incoming zone (the from-zone)
- ▶ An outgoing zone (the to-zone)
- ▶ An ordered set of policies between the from-zone and to-zone

Each policy consists of:

- ▶ A unique name for the policy.
- ▶ A set of match criteria defining the conditions that must be satisfied to apply the policy rule. The match criteria are based on a source IP address, destination IP address, and applications.
- ▶ A set of actions to be performed in case of a match - permit, deny, or reject.
- ▶ Accounting and auditing elements - counting, logging, or structured system logging.

Security policies configuration

To configure a match condition from a security policy, use the command:

```
{primary:node0}[edit]
ibm@J58S-1# set security policies from-zone from-zone to-zone to-zone policy
policy_name match match_statement
```

In the command:

- ▶ From-zone is the source zone
- ▶ To-zone is the destination zone
- ▶ Policy_name is the name of this policy
- ▶ Match_statement is the condition for selecting this policy

The match_statement can be one of source-address, destination-address, and application, with options.

To configure an action in a security policy, use the command

```
{primary:node0}[edit]
ibm@J58S-1# set security policies from-zone from-zone to-zone to-zone policy
policy_name then action
```

In the command:

- ▶ From-zone is the source zone
- ▶ To-zone is the destination zone
- ▶ Policy_name is the name of this policy
- ▶ Action is the action of this policy that can take values from permit, deny, reject, count and log

To verify the configuration of the security policies, use the command:

```
{primary:node0}[edit]
ibm@J58S-1# show security policies
```

To verify the status of security policies, use the operational command:

```
{primary:node0}  
ibm@J58S-1> show security policies [detail]
```

8.4.2 Security Policy Schedulers

This section presents an overview of Security Policy Schedulers and the commands to configure them.

Security Policy Schedulers overview

Schedulers are powerful features that allow a policy to be activated for a specified duration. You can define schedulers for a single (nonrecurrent) or recurrent time slot within which a policy is active. You can create schedulers irrespective of a policy, meaning that a scheduler cannot be used by any policies. However, if you want a policy to be active within a scheduled time, then you must first create a scheduler.

When a scheduler times out, the associated policy is deactivated and all sessions associated with the policy are also timed out.

If a policy contains a reference to a scheduler, the schedule determines when the policy is active, that is, when it can be used as a possible match for traffic. Schedulers allow you to restrict access to a resource for a period of time or remove a restriction.

The following guidelines apply to schedulers:

- ▶ A scheduler can have multiple policies associated with it; however, a policy cannot be associated with multiple schedulers.
- ▶ A policy is active during the time when the scheduler it refers to is also active.
- ▶ When a scheduler is off, the policy is unavailable for policy lookup.
- ▶ A scheduler can be configured as one of the following:
 - Scheduler can be active for a single time slot, as specified by a start date and time and a stop date and time.
 - Scheduler can be active forever (recurrent), but as specified by the daily schedule. The schedule on a specific day (time slot) takes priority over the daily schedule.
 - Scheduler can be active within a time slot as specified by the weekday schedule.
 - Scheduler can have a combination of two time slots (daily and time slot).

To configure a scheduler, use the command:

```
{primary:node0}[edit]  
ibm@J58S-1# set schedulers scheduler scheduler_name options
```

In the command:

- ▶ **Scheduler_name** is the name of the created scheduler
- ▶ Options are: **daily**, **sunday**, **monday**, **tuesday**, **wednesday**, **thursday**, **friday**, **saturday**. Also the options can have suboptions, such as **exclude** and **start-time**, **stop-time**.

To associate the defined schedule to a policy, use the command:

```
{primary:node0}[edit]  
ibm@J58S-1# set security policies from-zone from-zone to-zone to-zone policy  
policy_name scheduler-name scheduler_name
```

In the command:

- ▶ From-zone is the source zone
- ▶ To-zone is the destination zone
- ▶ Policy_name is the name of this policy
- ▶ Scheduler_name is the name of the scheduler to be associated with the policy

To verify the configuration of the schedulers, use the command:

```
{primary:node0}[edit]
ibm@J58S-1# show security schedulers
```

To verify the status of the schedulers, use the operational command:

```
{primary:node0}
ibm@J58S-1> show security schedulers
```

8.4.3 Security Policy Applications

This section presents an overview of Security Policy Applications and the commands to configure them.

Security Policy Applications overview

Applications are types of traffic for which protocol standards exist. Each application has a transport protocol and destination port number(s) associated with it, such as TCP/port 21 for FTP and TCP/port 23 for Telnet. When you create a policy, you must specify an application for it.

You can select one of the predefined applications from the application book, or a custom application or application set that you created. You can see which application you can use in a policy by using the show application CLI command.

Each predefined application has a source port range of 1–65535, which includes the entire set of valid port numbers. This prevents potential attackers from gaining access by using a source port outside of the range. If you need to use a different source port range for any predefined application, create a custom application.

Policy Application Sets overview

When you create a policy, you must specify an application, or service, for it to indicate that the policy applies to traffic of that type. Sometimes the same applications or a subset of them can be present in multiple policies, making it difficult to manage. Junos software allows you to create groups of applications called application sets. Application sets simplify the process by allowing you to manage a small number of application sets, rather than a large number of individual application entries.

The application (or application set) is referred to by security policies as match criteria for packets initiating sessions. If the packet matches the application type specified by the policy and all other criteria match, then the policy action is applied to the packet.

You can specify the name of an application set in a policy. In this case, if all of the other criteria match, any one of the applications in the application set serves as valid matching criteria; any is the default application name that indicates all possible applications.

Applications are created in the .../applications/application/<application-name> directory. You do not need to configure an application for any of the services that are predefined by the system.

In addition to predefined services, you can configure a custom service. After you create a custom service, you can refer to it in a policy.

Rather than create or add multiple individual application names to a policy, you can create an application set and refer to the name of the set in a policy. For example, for a group of employees, you can create an application set that contains all the approved applications.

To configure application sets, use the command:

```
{primary:node0}[edit]
ibm@J58S-1# set applications application-set application-set application
application
```

In the command:

- ▶ Application-set is the name of the application set
- ▶ Application is the name of the application assigned to application set

Custom application mappings

The application option specifies the Layer 7 application that maps to the Layer 4 application that you reference in a policy. A predefined application already has a mapping to a Layer 7 application. However, for custom applications, you must link the application to an application explicitly, especially if you want the policy to apply an Application Layer Gateway (ALG) or deep inspection to the custom application.

Junos software supports ALGs for numerous applications, including DNS, FTP, H.323, HTTP, RSH, SIP, Telnet, and TFTP.

Applying an ALG to a custom application involves the following two steps:

- ▶ Define a custom application with a name, timeout value, transport protocol, and source and destination ports.
- ▶ When configuring a policy, reference that application and the application type for the ALG that you want to apply.

To configure a custom TCP application, use the command:

```
{primary:node0}[edit]
ibm@J58S-1# set applications application custom-app-name protocol tcp [source-port
src-ports] [destination-port dst-ports] [inactivity-timeout timeout]
```

In the command:

- ▶ Custom-app-name is the name of the custom application
- ▶ Src-ports is the source port or range of source ports used by this custom application
- ▶ Dst-ports is the destination port or range of destination ports used by this custom application
- ▶ Timeout is the inactivity timeout for the application

The default timeout value of a custom application is 180 minutes. If you do not want an application to time out, type **never**.

To configure a custom UDP application, use the command:

```
{primary:node0}[edit]
ibm@J58S-1# set applications application custom-app-name protocol udp
[source-port src-ports] [destination-port dst-ports] [inactivity-timeout timeout]
```

In the command:

- Custom-app-name is the name of the custom application
 - Src-ports is the source port or range of source ports used by this custom application
 - Dst-ports is the destination port or range of destination ports used by this custom application
 - Timeout is the inactivity timeout for the application
- The default timeout value of a custom application is 180 minutes. If you do not want an application to time out, type **never**.

For other types of application, refer to the *JUNOS Software Security Configuration Guide, Release 10.1* available at <http://www.juniper.net/techpubs/software/junos-security/junos-security10.1/junos-security-swconfig-security/junos-security-swconfig-security.pdf>.

Understanding policy application timeout configuration and lookup

The application timeout value you set for an application determines the session timeout. You can set the timeout threshold for a predefined or custom application; you can use the application default timeout, specify a custom timeout, or use no timeout at all. Application timeout behavior is the same in virtual systems (vsys) security domains as at the root level.

Application timeout values are stored in the application entry database and in the corresponding vsys TCP and UDP port-based timeout tables. When you set a application timeout value, Junos software updates these tables with the new value. There are also default timeout values in the applications entry database, which are taken from predefined applications. You can set a timeout, but you cannot alter the default values.

Applications with multiple rule entries share the same timeout value. If multiple applications share the same protocol and destination port range, all applications share the last timeout value configured.

For single application entries, an application timeout lookup proceeds as follows:

- The specified timeout in the application entry database, if set.
- The default timeout in the application entry database, if specified in the predefined application.
- The protocol-based default timeout table. See Table 8-2.

Table 8-2 Default protocol timeout

Protocol	Default timeout (minutes)
TCP	30
UDP	1
ICMP	1
OSPF	1
Other	30

For application groups, including hidden groups created in multicell policy configurations, and for the predefined application ANY (if timeout is not set), application timeout lookup proceeds as follows:

- ▶ The vsys TCP and UDP port-based timeout table, if a timeout is set.
- ▶ The protocol-based default timeout table.

8.4.4 Security policy configuration example

Example 8-3 presents configuration for security policies with the following characteristics:

- ▶ We have two security zones, *inside* and *outside*, with characteristics defined in Example 8-2 on page 201.
- ▶ The security policy for traffic from zone inside to zone outside is:
 - Use policy name of in-out-1 and in-out-2
 - Permit and count traffic to out-mail address-set defined in Example 8-2 on page 201
 - Permit and count all traffic
- ▶ The security policy for traffic from zone outside to zone inside is:
 - Use policy name of out-in-1
 - Permit, count, and log ping, http, and https traffic to internal web servers
 - Internal web servers IP addresses are 192.168.1.101 and 192.168.1.102
 - Use application-set web to group the web services, and use junos-icmp-ping for ping
 - Use address-set web-srv for grouping the web servers
 - All other traffic must be handled by default deny action of the policy
- ▶ The security policy for traffic from zone inside to zone inside is:
 - Use policy name of in-in-1
 - Permit and count all traffic
- ▶ The security policy for traffic from zone outside to zone outside is:
 - Use policy name of out-out-1
 - Deny and count all traffic

Example 8-3 Security policy configuration

```
{primary:node0}[edit]
ibm@J58S-1# edit security policies from-zone inside to-zone outside

{primary:node0}[edit security policies from-zone inside to-zone outside]
ibm@J58S-1# set policy in-out-1 match source-address any

{primary:node0}[edit security policies from-zone inside to-zone outside]
ibm@J58S-1# set policy in-out-1 match destination-address out-mail

{primary:node0}[edit security policies from-zone inside to-zone outside]
ibm@J58S-1# set policy in-out-1 match application any

{primary:node0}[edit security policies from-zone inside to-zone outside]
ibm@J58S-1# set policy in-out-1 then count

{primary:node0}[edit security policies from-zone inside to-zone outside]
ibm@J58S-1# set policy in-out-1 then permit

{primary:node0}[edit security policies from-zone inside to-zone outside]
ibm@J58S-1# set policy in-out-2 match source-address any
```



```

{primary:node0}[edit security policies from-zone inside to-zone outside]
ibm@J58S-1# set policy in-out-2 match destination-address any

{primary:node0}[edit security policies from-zone inside to-zone outside]
ibm@J58S-1# set policy in-out-2 match application any

{primary:node0}[edit security policies from-zone inside to-zone outside]
ibm@J58S-1# set policy in-out-2 then permit

{primary:node0}[edit security policies from-zone inside to-zone outside]
ibm@J58S-1# top

{primary:node0}[edit]
ibm@J58S-1# set applications application-set web application junos-http

{primary:node0}[edit]
ibm@J58S-1# set applications application-set web application junos-https

{primary:node0}[edit]
ibm@J58S-1# set security zones security-zone inside address-book address web1
192.168.1.101

{primary:node0}[edit]
ibm@J58S-1# set security zones security-zone inside address-book address web2
192.168.1.102

{primary:node0}[edit]
ibm@J58S-1# set security zones security-zone inside address-book address-set
web-srv address web1

{primary:node0}[edit]
ibm@J58S-1# set security zones security-zone inside address-book address-set
web-srv address web2

{primary:node0}[edit]
ibm@J58S-1# edit security policies from-zone outside to-zone inside

{primary:node0}[edit security policies from-zone outside to-zone inside]
ibm@J58S-1# set policy out-in-1 match source-address any

{primary:node0}[edit security policies from-zone outside to-zone inside]
ibm@J58S-1# set policy out-in-1 match destination-address web-srv

{primary:node0}[edit security policies from-zone outside to-zone inside]
ibm@J58S-1# set policy out-in-1 match application web

{primary:node0}[edit security policies from-zone outside to-zone inside]
ibm@J58S-1# set policy out-in-1 match application junos-icmp-ping

{primary:node0}[edit security policies from-zone outside to-zone inside]
ibm@J58S-1# set policy out-in-1 then permit

{primary:node0}[edit security policies from-zone outside to-zone inside]
ibm@J58S-1# set policy out-in-1 then count

```

```

{primary:node0}[edit security policies from-zone outside to-zone inside]
ibm@J58S-1# set policy out-in-1 then log session-init session-close

{primary:node0}[edit security policies from-zone outside to-zone inside]
ibm@J58S-1# top edit security policies from-zone inside to-zone inside

{primary:node0}[edit security policies from-zone inside to-zone inside]
ibm@J58S-1# set policy in-in-1 match destination-address any

{primary:node0}[edit security policies from-zone inside to-zone inside]
ibm@J58S-1# set policy in-in-1 match source-address any

{primary:node0}[edit security policies from-zone inside to-zone inside]
ibm@J58S-1# set policy in-in-1 match application any

{primary:node0}[edit security policies from-zone inside to-zone inside]
ibm@J58S-1# set policy in-in-1 then permit

{primary:node0}[edit security policies from-zone inside to-zone inside]
ibm@J58S-1# set policy in-in-1 then count

{primary:node0}[edit security policies from-zone inside to-zone inside]
ibm@J58S-1# top edit security policies from-zone outside to-zone outside

{primary:node0}[edit security policies from-zone outside to-zone outside]
ibm@J58S-1# set policy out-out-1 match destination-address any

{primary:node0}[edit security policies from-zone outside to-zone outside]
ibm@J58S-1# set policy out-out-1 match source-address any

{primary:node0}[edit security policies from-zone outside to-zone outside]
ibm@J58S-1# set policy out-out-1 match application any

{primary:node0}[edit security policies from-zone outside to-zone outside]
ibm@J58S-1# set policy out-out-1 then count

{primary:node0}[edit security policies from-zone outside to-zone outside]
ibm@J58S-1# set policy out-out-1 then deny

{primary:node0}[edit security policies from-zone outside to-zone outside]
ibm@J58S-1# top

{primary:node0}[edit]
ibm@J58S-1# show security policies
from-zone inside to-zone outside {
    policy in-out-1 {
        match {
            source-address any;
            destination-address out-mail;
            application any;
        }
        then {
            permit;
            count;

```

```

    }
  }
  policy in-out-2 {
    match {
      source-address any;
      destination-address any;
      application any;
    }
    then {
      permit;
    }
  }
}
from-zone outside to-zone inside {
  policy out-in-1 {
    match {
      source-address any;
      destination-address web-srv;
      application [ web junos-icmp-ping ];
    }
    then {
      permit;
      log {
        session-init;
        session-close;
      }
      count;
    }
  }
}
from-zone inside to-zone inside {
  policy in-in-1 {
    match {
      source-address any;
      destination-address any;
      application any;
    }
    then {
      permit;
      count;
    }
  }
}
from-zone outside to-zone outside {
  policy out-out-1 {
    match {
      source-address any;
      destination-address any;
      application any;
    }
    then {
      deny;
      count;
    }
  }
}

```

```

}

{primary:node0}[edit]
ibm@J58S-1# commit
node0:
commit complete

{primary:node0}[edit]
ibm@J58S-1# run show security policies
node0:
-----
Default policy: deny-all
From zone: inside, To zone: outside
  Policy: in-out-1, State: enabled, Index: 8, Sequence number: 1
    Source addresses: any
    Destination addresses: out-mail
    Applications: any
    Action: permit, count
  Policy: in-out-2, State: enabled, Index: 9, Sequence number: 2
    Source addresses: any
    Destination addresses: any
    Applications: any
    Action: permit
From zone: inside, To zone: inside
  Policy: in-in-1, State: enabled, Index: 6, Sequence number: 1
    Source addresses: any
    Destination addresses: any
    Applications: any
    Action: permit, count
From zone: outside, To zone: inside
  Policy: out-in-1, State: enabled, Index: 5, Sequence number: 1
    Source addresses: any
    Destination addresses: web-srv
    Applications: web, junos-icmp-ping
    Action: permit, log, count
From zone: outside, To zone: outside
  Policy: out-out-1, State: enabled, Index: 7, Sequence number: 1
    Source addresses: any
    Destination addresses: any
    Applications: any
    Action: deny, count

{primary:node0}[edit]
ibm@J58S-1# run show security policies detail
node0:
-----
Default policy: deny-all
Policy: in-out-1, action-type: permit, State: enabled, Index: 8
  Policy Type: Configured
  Sequence number: 1
  From zone: inside, To zone: outside
  Source addresses:
    any: 0.0.0.0/0
  Destination addresses:
    out-imap: 10.10.10.102/32

```

```

    out-pop3: 10.10.10.101/32
    out-smtp: 10.10.10.100/32
Application: any
  IP protocol: 0, ALG: 0, Inactivity timeout: 0
    Source port range: [0-0]
    Destination port range: [0-0]
Policy statistics:
  Input bytes      :                1176          0 bps
  Output bytes     :                1176          0 bps
  Input packets    :                 14          0 pps
  Output packets   :                 14          0 pps
  Session rate     :                 14          0 sps
  Active sessions  :                  0
  Session deletions:                 14
  Policy lookups   :                 14
Policy: in-out-2, action-type: permit, State: enabled, Index: 9
  Policy Type: Configured
  Sequence number: 2
  From zone: inside, To zone: outside
  Source addresses:
    any: 0.0.0.0/0
  Destination addresses:
    any: 0.0.0.0/0
  Application: any
    IP protocol: 0, ALG: 0, Inactivity timeout: 0
      Source port range: [0-0]
      Destination port range: [0-0]
Policy: in-in-1, action-type: permit, State: enabled, Index: 6
  Policy Type: Configured
  Sequence number: 1
  From zone: inside, To zone: inside
  Source addresses:
    any: 0.0.0.0/0
  Destination addresses:
    any: 0.0.0.0/0
  Application: any
    IP protocol: 0, ALG: 0, Inactivity timeout: 0
      Source port range: [0-0]
      Destination port range: [0-0]
Policy statistics:
  Input bytes      :                  0          0 bps
  Output bytes     :                  0          0 bps
  Input packets    :                  0          0 pps
  Output packets   :                  0          0 pps
  Session rate     :                  0          0 sps
  Active sessions  :                  0
  Session deletions:                  0
  Policy lookups   :                  0
Policy: out-in-1, action-type: permit, State: enabled, Index: 5
  Policy Type: Configured
  Sequence number: 1
  From zone: outside, To zone: inside
  Source addresses:
    any: 0.0.0.0/0
  Destination addresses:

```

```

web2: 192.168.1.102/32
web1: 192.168.1.101/32
Application: web
  IP protocol: tcp, ALG: 0, Inactivity timeout: 1800
    Source port range: [0-0]
    Destination port range: [80-80]
  IP protocol: tcp, ALG: 0, Inactivity timeout: 1800
    Source port range: [0-0]
    Destination port range: [443-443]
Application: junos-icmp-ping
  IP protocol: icmp, ALG: 0, Inactivity timeout: 60
    ICMP Information: type=8, code=0
Session log: at-create, at-close
Policy statistics:
  Input bytes      :                9241          0 bps
  Output bytes     :                9241          0 bps
  Input packets    :                 95          0 pps
  Output packets   :                 95          0 pps
  Session rate     :                 26          0 sps
  Active sessions  :                  0
  Session deletions:                 26
  Policy lookups   :                 25
Policy: out-out-1, action-type: deny, State: enabled, Index: 7
Policy Type: Configured
Sequence number: 1
From zone: outside, To zone: outside
Source addresses:
  any: 0.0.0.0/0
Destination addresses:
  any: 0.0.0.0/0
Application: any
  IP protocol: 0, ALG: 0, Inactivity timeout: 0
    Source port range: [0-0]
    Destination port range: [0-0]
Policy statistics:
  Input bytes      :                  0          0 bps
  Output bytes     :                  0          0 bps
  Input packets    :                  0          0 pps
  Output packets   :                  0          0 pps
  Session rate     :                  0          0 sps
  Active sessions  :                  0
  Session deletions:                  0
  Policy lookups   :                  0

{primary:node0}[edit]
ibm@J58S-1#

```

8.5 Network Address Translation

This section presents an overview of Network Address Translation (NAT) and how to configure and monitor NAT features on IBM j-type security appliances.

8.5.1 NAT overview

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

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

The following types of NAT are supported on IBM j-type appliances:

- ▶ Static NAT
- ▶ Destination NAT
- ▶ Source NAT

NAT rule sets and rules

NAT processing centers on the evaluation of NAT *rule sets* and *rules*. A rule set determines the overall direction of the traffic to be processed. For example, a rule set can select traffic from a particular interface or to a specific zone. A rule set can contain multiple rules. After a rule set is found that matches specific traffic, each rule in the rule set is evaluated for a match. Each rule in the rule set further specifies the traffic to be matched and the action to be taken when traffic matches the rule.

NAT rule sets

A rule set specifies a general set of matching conditions for traffic. For static NAT and destination NAT, a rule set specifies one of the following:

- ▶ Source interface
- ▶ Source zone
- ▶ Source routing instance

For source NAT rule sets, you configure both source and destination conditions:

- ▶ Source interface, zone, or routing instance
- ▶ Destination interface, zone, or routing instance

It is possible for a packet to match more than one rule set; in this case, the rule set with the more specific match is used. An interface match is considered more specific than a zone match, which is more specific than a routing instance match. If a packet matches both a destination NAT rule set that specifies a source zone and a destination NAT rule set that specifies a source interface, the rule set that specifies the source interface is the more specific match.

Source NAT rule set matching is more complex because you specify both source and destination conditions in a source NAT rule set. In the case where a packet matches more than one source NAT rule set, the rule set chosen is based on the following source/destination conditions (in order of priority):

1. Source interface/destination interface
2. Source zone/destination interface
3. Source routing instance/destination interface
4. Source interface/destination zone
5. Source zone/destination zone
6. Source routing instance/destination zone
7. Source interface/destination routing instance
8. Source zone/destination routing instance
9. Source routing instance/destination routing instance

For example, you can configure rule set A, which specifies a source interface and a destination zone, and rule set B, which specifies a source zone and a destination interface. If a packet matches both rule sets, rule set B is the more specific match.

Source NAT rule sets: You cannot specify the same source and destination conditions for source NAT rule sets.

NAT rules

After a rule set that matches the traffic is found, each rule in the rule set is evaluated, in order, for a match. NAT rules can match on the following packet information:

- ▶ Destination address (for static NAT only)
- ▶ Source and destination address (for destination and source NAT)
- ▶ Destination port (for destination and source NAT)

The first rule in the rule set that matches the traffic is used. If a packet matches a rule in a rule set during session establishment, traffic is processed according to the action specified by that rule.

Rule processing

The NAT type determines the order in which NAT rules are processed. During the first packet processing for a flow, NAT rules are applied in the following order:

1. Static NAT rules
2. Destination NAT rules
3. Route lookup
4. Security policy lookup
5. Reverse mapping of static NAT rules
6. Source NAT rules

Static NAT and destination NAT rules are processed before route and security policy lookup. Static NAT rules take precedence over destination NAT rules. Reverse mapping of static NAT rules takes place after route and security policy lookup and takes precedence over source NAT rules. Source NAT rules are processed after route and security policy lookup and after reverse mapping of static NAT rules.

The configuration of rules and rule sets is basically the same for each type of NAT—source, destination, or static. But because both destination and static NAT are processed before route lookup, you cannot specify the destination zone, interface or routing instance in the rule set.

NAT proxy ARP

You use NAT proxy ARP functionality to configure proxy ARP entries for IP addresses that require either source or destination NAT and that are in the same subnet as the ingress interface.

The device performs proxy ARP for the following conditions:

- ▶ When addresses defined in the static NAT and source NAT pool are in the same subnet as that of the ingress interface
- ▶ When addresses in the original destination address entry in the destination NAT rules are in the same subnet as that of the ingress interface

8.5.2 Static NAT

Static NAT defines a one-to-one mapping from one IP subnet to another IP subnet. The mapping includes destination IP address translation in one direction and source IP address translation in the reverse direction. From the NAT device, the original destination address is the virtual host IP address while the mapped-to address is the real host IP address.

Static NAT allows connections to be originated from either side of the network, but translation is limited to one-to-one or between blocks of addresses of the same size.

For each private address, a public address must be allocated. No address pools are necessary.

Note: The original destination address, along with other addresses in source and destination NAT pools, must not overlap within the same routing instance.

Static NAT does not perform port address translation (PAT) and no address pools are needed for static NAT.

In NAT rule lookup, static NAT rules take precedence over destination NAT rules and reverse mapping of static NAT rules take precedence over source NAT rules.

Static NAT Rules

Static NAT rules specify two layers of match conditions:

- ▶ Traffic direction: Allows you to specify from interface, from zone, or from routing-instance.
- ▶ Packet information: Destination IP address.

If multiple static NAT rules overlap in the match conditions, the most specific rule is chosen. For example, if rules A and B specify the same source and destination IP addresses, but rule A specifies traffic from zone 1 and rule B specifies traffic from interface ge-0/0/0, rule B is used to perform static NAT. An interface match is considered to be more specific than a zone match, which is more specific than a routing instance match.

For the static NAT rule action, specify the translated address and (optionally) the routing instance.

In NAT lookup, static NAT rules take precedence over destination NAT rules and reverse mapping of static NAT rules takes precedence over source NAT rules.

Static NAT Configuration

1. Configure static NAT rules that align with your network and security requirements.
2. Configure NAT proxy ARP entries for IP addresses in the same subnet of the ingress interface.

To configure static NAT rules, use the following commands.

- ▶ For specifying the source of packets for NAT:

```
{primary:node0}[edit]
ibm@J58S-1# set security nat static rule-set rule_set_name from source
```

In the command:

- Rule_set_name is the name of the rule set
- Source is the source for which we need NAT

The source can be **zone**, **interface**, or **routing-instance** level.

- For defining the matching packets destination address that are subject of this NAT rule:

```
{primary:node0}[edit]
ibm@J58S-1# set security nat static rule-set rule_set_name rule rule_name match
destination-address IP_address
```

In the command:

- Rule_set_name is the name of the rule set
- Rule_name is the name of the rule
- IP_address is the global IP address for the NAT

- For defining the static IP address with which we translate the destination IP address of the matching traffic:

```
{primary:node0}[edit]
ibm@J58S-1# set security nat static rule-set rule_set_name rule rule_name then
static-nat prefix IP_address
```

In the command:

- Rule_set_name is the name of the rule set
- Rule_name is the name of the rule
- IP_address is the static IP address used for translation (can also be an IP prefix for multiple addresses)

To configure NAT proxy ARP if necessary, use the command:

```
{primary:node0}[edit]
ibm@J58S-1# set security nat proxy-arp interface interface address IP_address
```

In the command:

- Interface is the name of the interface in type-fpc/pic/port.logical-unit-number notation
- IP_address is the IP address for which to perform proxy arp

To verify static nat configuration, use the command:

```
{primary:node0}[edit]
ibm@J58S-1# show security nat static
```

To verify static nat operation, use the command:

```
{primary:node0}
ibm@J58S-1> show security nat static rule all
```

8.5.3 Static NAT configuration example

In Example 8-4 on page 223 we used the security zone definition from Example 8-2 on page 201 and security policy definition from Example 8-3 on page 212, and based on this preconfiguration we demonstrate two static NAT translations with the following characteristics:

- Define a static rule-set named **rule-set1** from zone outside
- Define a static NAT rule named **rule1** that translates the destination IP of all traffic to 192.168.10.101 with the static IP of 192.168.1.101
- Define a static NAT rule named **rule2** that translates the destination IP of all traffic to 10.10.10.11 with the static IP of 192.168.1.102
- For proper operation of this setup, configure proxy arp on interface ge-1/1/0.0 for the IP address 10.10.10.11

Example 8-4 Static NAT configuration

```
{primary:node0}[edit]
ibm@J58S-1# set security nat static rule-set rule_set1 from zone outside

{primary:node0}[edit]
ibm@J58S-1# set security nat static rule-set rule_set1 rule rule1 match
destination-address 192.168.10.101

{primary:node0}[edit]
ibm@J58S-1# set security nat static rule-set rule_set1 rule rule1 then static-nat
prefix 192.168.1.101

{primary:node0}[edit]
ibm@J58S-1# edit security nat static rule-set rule_set1

{primary:node0}[edit security nat static rule-set rule_set1]
ibm@J58S-1# set rule rule2 match destination-address 10.10.10.11

{primary:node0}[edit security nat static rule-set rule_set1]
ibm@J58S-1# set rule rule2 then static-nat prefix 192.168.1.102

{primary:node0}[edit security nat static rule-set rule_set1]
ibm@J58S-1# top

{primary:node0}[edit]
ibm@J58S-1# set security nat proxy-arp interface ge-1/1/0.0 address 10.10.10.11

{primary:node0}[edit]
ibm@J58S-1# commit
node0:
configuration check succeeds
node1:
commit complete
node0:
commit complete

{primary:node0}[edit]
ibm@J58S-1# show security nat
static {
    rule-set rule_set1 {
        from zone outside;
        rule rule1 {
            match {
                destination-address 192.168.10.101/32;
            }
            then {
                static-nat prefix 192.168.1.101/32;
            }
        }
        rule rule2 {
            match {
                destination-address 10.10.10.11/32;
            }
            then {
                static-nat prefix 192.168.1.102/32;
            }
        }
    }
}
```

```

    }
  }
}
proxy-arp {
  interface ge-1/1/0.0 {
    address {
      10.10.10.11/32;
    }
  }
}

```

```

{primary:node0}[edit]
ibm@J58S-1# run show security nat static rule all

```

```

node0:
-----
Total static-nat rules: 2

Static NAT rule: rule2                                Rule-set: rule_set1
  Rule-Id                                           : 2
  Rule position                                     : 5
  From zone                                         : outside
  Destination addresses                             : 10.10.10.11
  Host addresses                                    : 192.168.1.102
  Netmask                                           : 255.255.255.255
  Host routing-instance                             : N/A
  Translation hits                                  : 4 <----- rule hitcount

Static NAT rule: rule1                                Rule-set: rule_set1
  Rule-Id                                           : 1
  Rule position                                     : 4
  From zone                                         : outside
  Destination addresses                             : 192.168.10.101
  Host addresses                                    : 192.168.1.101
  Netmask                                           : 255.255.255.255
  Host routing-instance                             : N/A
  Translation hits                                  : 2 <----- rule hitcount

```

```

node1:
-----
Total static-nat rules: 2

Static NAT rule: rule2                                Rule-set: rule_set1
  Rule-Id                                           : 2
  Rule position                                     : 5
  From zone                                         : outside
  Destination addresses                             : 10.10.10.11
  Host addresses                                    : 192.168.1.102
  Netmask                                           : 255.255.255.255
  Host routing-instance                             : N/A
  Translation hits                                  : 0

Static NAT rule: rule1                                Rule-set: rule_set1
  Rule-Id                                           : 1

```

```
Rule position          : 4
From zone              : outside
Destination addresses  : 192.168.10.101
Host addresses         : 192.168.1.101
Netmask               : 255.255.255.255
Host routing-instance : N/A
Translation hits       : 0
```

```
{primary:node0}[edit]
ibm@J58S-1#
```

8.5.4 Destination NAT

Destination NAT is the translation of the destination IP address of a packet entering the IBM j-type security appliance. Destination NAT is used to redirect traffic destined to a virtual host (identified by the original destination IP address) to the real host (identified by the translated destination IP address).

Note: When destination NAT is performed, the destination IP address is translated according to configured destination NAT rules and then security policies are applied.

Destination NAT allows connections to be initiated only for incoming network connections—for example, from the Internet to a private network. Destination NAT is commonly used to perform the following actions:

- ▶ Translate a single IP address to another address (for example, to allow a device on the Internet to connect to a host on a private network).
- ▶ Translate a contiguous block of addresses to another block of addresses of the same size (for example, to allow access to a group of servers).
- ▶ Translate a destination IP address and port to another destination IP address and port (for example, to allow access to multiple services using the same IP address but different ports).

The following types of destination NAT are supported:

- ▶ Translation of the original destination IP address to an IP address from a user-defined pool. This type of translation does not include Port Address Translation (PAT). If the original destination IP address range is larger than the address range in the user-defined address pool, any untranslated packets are dropped.
- ▶ Translation of the original destination IP address (and optional port number) to one specific IP address (and port number) from a user-defined pool.

Destination NAT address pools

For destination NAT address pools, specify the following:

- ▶ Name of the destination NAT address pool
- ▶ Destination address or address range

Note: Do not overlap NAT addresses for source NAT, destination NAT, and static NAT within one routing instance.

- ▶ Destination port that is used for port forwarding
- ▶ Routing instance to which the pool belongs (the default is the main inet.0 routing instance)

Destination NAT Rules

Destination NAT rules specify two layers of match conditions:

- ▶ Traffic direction: Allows you to specify from interface, from zone, or from routing-instance.
- ▶ Packet information: Can be source IP addresses, destination IP address or subnet, or a single destination port number.

If multiple destination NAT rules overlap in the match conditions, the most specific rule is chosen. For example, if rules A and B specify the same source and destination IP addresses, but rule A specifies traffic from zone 1 and rule B specifies traffic from interface ge-0/0/0, rule B is used to perform destination NAT. An interface match is considered to be more specific than a zone match, which is more specific than a routing instance match.

The actions you can specify for a destination NAT rule are:

- ▶ Off: Do not perform destination NAT.
- ▶ Pool: Use the specified user-defined address pool to perform destination NAT.

Destination NAT rules are applied to traffic in the first packet that is processed for the flow or in the fast path for the ALG. Destination NAT rules are processed after static NAT rules but before source NAT rules.

Destination NAT Configuration

The main configuration tasks for destination NAT are as follows:

1. Configure a destination NAT address pool that aligns with your network and security requirements.
2. Configure destination NAT rules that align with your network and security requirements.
3. Configure NAT proxy ARP entries for IP addresses in the same subnet of the ingress interface.

To configure a destination NAT address pool, use the command:

```
{primary:node0}[edit]
ibm@J58S-1# set security nat destination pool pool_name address first_IP_address
to last_IP_address
```

In the command:

- ▶ Pool_name is the name of the destination address pool we want to create
- ▶ First_IP_address is the first IP address in the pool, and last_IP_address is the last IP address in the pool. If you do not specify the last IP address, only one address will be added to the pool. You can also use a mask to the first_IP_address to specify multiple IP addresses to the pool.

To configure destination NAT rules, use the commands:

- ▶ For specifying the source of packets for NAT:

```
{primary:node0}[edit]
ibm@J58S-1# set security nat destination rule-set rule_set_name from source
```

In the command:

- Rule_set_name is the name of the rule set
- Source is the source for which we need NAT. The source can be zone, interface, or routing-instance level.

- For defining the matching packets destination address that are subject of this NAT rule:

```
{primary:node0}[edit]
ibm@J58S-1# set security nat destination rule-set rule_set_name rule rule_name
match destination-address IP_address
```

In the command:

- *Rule_set_name* is the name of the rule set
- *Rule_name* is the name of the rule
- *IP_address* is the global destination IP address for the NAT

- For defining the address pool from which we translate the destination IP address of the traffic:

```
{primary:node0}[edit]
ibm@J58S-1# set security nat destination rule-set rule_set_name rule rule_name
then destination-nat pool pool_name
```

In the command:

- *Rule_set_name* is the name of the rule set
- *Rule_name* is the name of the rule
- *Pool_name* is the name of the destination address pool we want to use for this rule

To configure NAT proxy ARP if necessary, use the command:

```
{primary:node0}[edit]
ibm@J58S-1# set security nat proxy-arp interface interface address IP_address
```

In the command:

- *Interface* is the name of the interface in type-fpc/pic/port.logical-unit-number notation
- *IP_address* is the IP address for which to perform proxy arp

To verify destination nat configuration, use the command:

```
{primary:node0}[edit]
ibm@J58S-1# show security nat destination
```

To verify destination nat operation, use the command:

```
{primary:node0}
ibm@J58S-1> show security nat destination rule all
```

To verify destination pool operation for nat, use the command:

```
{primary:node0}
ibm@J58S-1> show security nat destination pool all
```

To view a summary of destination nat operation, use the command:

```
{primary:node0}
ibm@J58S-1> show security nat destination summary
```

8.5.5 Destination NAT configuration example

In Example 8-5 on page 228, we used the security zone definition from Example 8-2 on page 201 and security policy definition from Example 8-3 on page 212, and based on this preconfiguration we demonstrate three destination NAT translations with the following characteristics:

- Define a destination rule-set named **rule-set1** from zone outside

- Define a destination address pool **dpool1** with IP of 192.168.1.101
- Define a destination address pool **dpool2** with IP of 192.168.1.102 to 192.168.1.105
- Define a destination NAT rule named **rule1** that translates the destination IP of all traffic to 192.168.10.101 with the IP from destination pool **dpool1**
- Define a destination NAT rule named **rule2** that translates the destination IP of all traffic to 192.168.20.102 with the IP from destination pool **dpool2**
- Define a destination NAT rule named **rule3** that translates the destination IP of all traffic to 192.168.30.103 with the IP from destination pool **dpool2**

Example 8-5 Destination NAT configuration

```
{primary:node0}[edit]
ibm@J58S-1# set security nat destination pool dpool-1 address 192.168.1.101

{primary:node0}[edit]
ibm@J58S-1# set security nat destination rule-set rule-set1 from zone outside

{primary:node0}[edit]
ibm@J58S-1# set security nat destination rule-set rule-set1 rule rule1 match
destination-address 192.168.10.101

{primary:node0}[edit]
ibm@J58S-1# set security nat destination rule-set rule-set1 rule rule1 then
destination-nat pool dpool-1

{primary:node0}[edit]
ibm@J58S-1# set security nat destination pool dpool-2 address 192.168.1.102 to
192.168.1.105

{primary:node0}[edit]
ibm@J58S-1# edit security nat destination rule-set rule-set1

{primary:node0}[edit security nat destination rule-set rule-set1]
ibm@J58S-1# set rule rule2 match destination-address 192.168.20.102

{primary:node0}[edit security nat destination rule-set rule-set1]
ibm@J58S-1# set rule rule2 then destination-nat pool dpool-2

{primary:node0}[edit security nat destination rule-set rule-set1]
ibm@J58S-1# set rule rule3 match destination-address 192.168.30.103

{primary:node0}[edit security nat destination rule-set rule-set1]
ibm@J58S-1# set rule rule3 then destination-nat pool dpool-2

{primary:node0}[edit security nat destination rule-set rule-set1]
ibm@J58S-1# top

{primary:node0}[edit]
bm@J58S-1# show security nat destination
pool dpool-1 {
    address 192.168.1.101/32;
}
pool dpool-2 {
    address 192.168.1.102/32 to 192.168.1.105/32;
```



```

}
rule-set rule-set1 {
  from zone outside;
  rule rule1 {
    match {
      destination-address 192.168.10.101/32;
    }
    then {
      destination-nat pool dpool-1;
    }
  }
  rule rule2 {
    match {
      destination-address 192.168.20.102/32;
    }
    then {
      destination-nat pool dpool-2;
    }
  }
  rule rule3 {
    match {
      destination-address 192.168.30.103/32;
    }
    then {
      destination-nat pool dpool-2;
    }
  }
}
}

```

```

{primary:node0}[edit]
ibm@J58S-1# commit
node0:
configuration check succeeds
node1:
commit complete
node0:
commit complete

```

```

{primary:node0}[edit]
ibm@J58S-1# run show security nat destination rule all
node0:

```

```

-----
Total destination-nat rules: 3

```

```

Destination NAT rule: rule1          Rule-set: rule-set1
  Rule-Id                          : 1
  Rule position                     : 1
  From zone                         : outside
  Destination addresses              : 192.168.10.101 - 192.168.10.101
  Action                            : dpool-1
  Destination port                   : 0
  Translation hits                   : 16

```

```

Destination NAT rule: rule2          Rule-set: rule-set1
  Rule-Id                          : 2

```

```

Rule position      : 2
From zone          : outside
  Destination addresses : 192.168.20.102 - 192.168.20.102
Action             : dpool-2
Destination port    : 0
Translation hits    : 3

```

```

Destination NAT rule: rule3          Rule-set: rule-set1
Rule-Id                             : 3
Rule position                         : 3
From zone                           : outside
  Destination addresses               : 192.168.30.103 - 192.168.30.103
Action                               : dpool-2
Destination port                      : 0
Translation hits                     : 4

```

node1:

Total destination-nat rules: 3

```

Destination NAT rule: rule1          Rule-set: rule-set1
Rule-Id                             : 1
Rule position                         : 1
From zone                           : outside
  Destination addresses               : 192.168.10.101 - 192.168.10.101
Action                               : dpool-1
Destination port                      : 0
Translation hits                     : 0

```

```

Destination NAT rule: rule2          Rule-set: rule-set1
Rule-Id                             : 2
Rule position                         : 2
From zone                           : outside
  Destination addresses               : 192.168.20.102 - 192.168.20.102
Action                               : dpool-2
Destination port                      : 0
Translation hits                     : 0

```

```

Destination NAT rule: rule3          Rule-set: rule-set1
Rule-Id                             : 3
Rule position                         : 3
From zone                           : outside
  Destination addresses               : 192.168.30.103 - 192.168.30.103
Action                               : dpool-2
Destination port                      : 0
Translation hits                     : 0

```

{primary:node0}[edit]

ibm@J58S-1# **run show security nat destination pool all**

node0:

Total destination-nat pools: 2

```

Pool name      : dpool-1
Pool id        : 1

```

```

Routing instance: default
Total address   : 1
Translation hits: 16
Address range   Port
192.168.1.101 - 192.168.1.101 0

```

```

Pool name       : dpool-2
Pool id         : 2
Routing instance: default
Total address   : 4
Translation hits: 17
Address range   Port
192.168.1.102 - 192.168.1.105 0

```

node1:

Total destination-nat pools: 2

```

Pool name       : dpool-1
Pool id         : 1
Routing instance: default
Total address   : 1
Translation hits: 0
Address range   Port
192.168.1.101 - 192.168.1.101 0

```

```

Pool name       : dpool-2
Pool id         : 2
Routing instance: default
Total address   : 4
Translation hits: 0
Address range   Port
192.168.1.102 - 192.168.1.105 0

```

{primary:node0}[edit]

ibm@J58S-1# **run show security nat destination summary**

node0:

Total pools: 2

Pool name	Address Range	Routing Instance	Port	Total
Address				
dpool-1	192.168.1.101 - 192.168.1.101	default	0	1
dpool-2	192.168.1.102 - 192.168.1.105	default	0	4

Total rules: 3

Rule name	Rule set	From	Action
rule1	rule-set1	outside	dpool-1
rule2	rule-set1	outside	dpool-2
rule3	rule-set1	outside	dpool-2

node1:

Total pools: 2

Pool name	Address	Routing	Port	Total
-----------	---------	---------	------	-------

Range		Instance		
Address				
dpool-1	192.168.1.101 - 192.168.1.101	default	0	1
dpool-2	192.168.1.102 - 192.168.1.105	default	0	4
Total rules: 3				
Rule name	Rule set	From	Action	
rule1	rule-set1	outside	dpool-1	
rule2	rule-set1	outside	dpool-2	
rule3	rule-set1	outside	dpool-2	
{primary:node0}[edit]				
ibm@J58S-1#				

8.5.6 Source NAT

Source NAT is the translation of the source IP address of a packet leaving the IBM j-type security appliance. Source NAT is used to allow hosts with private IP addresses to access a public network.

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

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

Translation to the address of the egress interface does not require an address pool; all other source NAT translations require configuration of an address pool. One-to-one and many-to-many translations for address blocks of the same size do not require port translation because there is an available address in the pool for every address to be translated.

If the size of the address pool is smaller than the number of addresses to be translated, either the total number of concurrent addresses that can be translated is limited by the size of the address pool or port translation must be used. For example, if a block of 253 addresses is translated to an address pool of 10 addresses, a maximum of 10 devices can be connected concurrently unless port translation is used.

The following types of source NAT are supported:

- ▶ Translation of the original source IP address to the egress interface's IP address (also called interface NAT). Port address translation is always performed.
- ▶ Translation of the original source IP address to an IP address from a user-defined address pool without port address translation. The association between the original source IP address to the translated source IP address is dynamic. However, after there is an association, the same association is used for the same original source IP address for new traffic that matches the same NAT rule.

- ▶ Translation of the original source IP address to an IP address from a user-defined address pool with port address translation. The association between the original source IP address to the translated source IP address is dynamic. Even if an association exists, the same original source IP address might be translated to a different address for new traffic that matches the same NAT rule.
- ▶ Translation of the original source IP address to an IP address from a user-defined address pool by shifting the IP addresses. This type of translation is one-to-one, static, and without port address translation. If the original source IP address range is larger than the IP address range in the user-defined pool, untranslated packets are dropped.

Source NAT Pools

For source NAT address pools, specify the following:

- ▶ Name of the source NAT address pool.
- ▶ Up to eight address or address ranges.

Note: Do not overlap NAT addresses for source NAT, destination NAT, and static NAT within one routing instance.

- ▶ Routing instance to which the pool belongs (the default is the main inet.0 routing instance).
- ▶ No port translation (optional): By default, port address translation is performed with source NAT. If you specify the port no-translation option, the number of hosts that the source NAT pool can support is limited to the number of addresses in the pool.
- ▶ Overflow pool (optional): Packets are dropped if there are no addresses available in the designated source NAT pool. To prevent that from happening when the port no-translation option is configured, you can specify an overflow pool. After addresses from the original source NAT pool are exhausted, IP addresses and port numbers are allocated from the overflow pool. A user-defined source NAT pool or an egress interface can be used as the overflow pool. (When the overflow pool is used, the pool ID is returned with the address.)
- ▶ IP address shifting (optional): A range of original source IP addresses can be mapped to another range of IP addresses by shifting the IP addresses. Specify the host-address-base option with the base address of the original source IP address range.

When the raise-threshold option is configured for source NAT, an SNMP trap is triggered if the source NAT pool utilization rises above this threshold. If the optional clear-threshold option is configured, an SNMP trap is triggered if the source NAT pool utilization drops below this threshold. If clear-threshold is not configured it is set by default to 80 percent of the raise-threshold value.

Source NAT Pools with PAT

Using the source pool with Port Address Translation (PAT), Junos software translates both the source IP address and the port number of the packets. When PAT is used, multiple hosts can share the same IP address.

Junos software maintains a list of assigned port numbers to distinguish what session belongs to which host. When PAT is enabled, up to 64,500 hosts can share a single IP address. Each source pool can contain multiple IP addresses, multiple IP address ranges, or both. For a source pool with PAT, Junos software might assign different addresses to a single host for different concurrent sessions, unless the source pool or Junos software has the persistent address feature enabled.

For interface source pool and source pool with PAT, range (1024, 65535) is available for port number mapping per IP address. Within range (1024, 63487) one port is allocated at a time. In range (63488, 65535), two ports are allocated at a time for RTP/RTCP applications such as SIP, H.323, and RTSP.

When a host initiates several sessions that match a policy that requires network address translation and is assigned an address from a source pool that has PAT enabled, the device assigns a different source IP address for each session. Such random address assignment can be problematic for services that create multiple sessions that require the same source IP address for each session. For example, it is important to have the same IP address for multiple sessions when using the AOL Instant Message (AIM) client.

To ensure that the router assigns the same IP address from a source pool to a host for multiple concurrent sessions, you can enable a persistent IP address per router.

Source NAT Pools Without PAT

When you define a source pool, Junos software enables PAT by default. To disable PAT, you must specify no port translation when you are defining a source pool.

When using a source pool without PAT, Junos software performs source Network Address Translation for the IP address without performing PAT for the source port number. For applications that require that a particular source port number remain fixed, you must use source pool without PAT.

The source pool can contain multiple IP addresses, multiple IP address ranges, or both. For source pool without PAT, Junos software assigns one translated source address to the same host for all its concurrent sessions.

Pool utilization for each source pool without PAT is computed. You can turn on pool utilization alarm by configuring alarm thresholds. An SNMP trap is triggered every time pool utilization rises above a threshold and goes below a threshold.

Source NAT Rules

Source NAT rules specify two layers of match conditions:

- ▶ Traffic direction: Allows you to specify combinations of from interface, from zone, or from routing-instance and to interface, to zone, or to routing-instance. You cannot configure the same from and to contexts for different rule sets.
- ▶ Packet information: Can be source and destination IP addresses or subnets.

If multiple source NAT rules overlap in the match conditions, the most specific rule is chosen. For example, if rules A and B specify the same source and destination IP addresses, but rule A specifies traffic from zone 1 to zone 2 and rule B specifies traffic from zone 1 to interface ge-0/0/0, rule B is used to perform source NAT. An interface match is considered to be more specific than a zone match, which is more specific than a routing instance match.

The actions you can specify for a source NAT rule are:

- ▶ Off: Do not perform source NAT.
- ▶ Pool: Use the specified user-defined address pool to perform source NAT.
- ▶ Interface: Use the egress interface's IP address to perform source NAT.

Source NAT rules are applied to traffic in the first packet that is processed for the flow or in the fast path for the ALG. Source NAT rules are processed after static NAT rules, destination NAT rules, and reverse mapping of static NAT rules and after route and security policy lookup.

Source NAT Configuration

The main configuration tasks for source NAT are:

1. Configure a source NAT address pool that aligns with your network and security requirements: Not needed for interface NAT.
2. Configure pool utilization alarms (optional): Specify thresholds for pool utilization.
3. Configure address persistent (optional): Ensures that the same IP address is assigned from the source NAT pool to a host for multiple concurrent sessions.
4. Configure source NAT rules that align with your network and security requirements.
5. Configure NAT proxy ARP entries for IP addresses in the same subnet of the ingress interface.

To configure a source NAT address pool, use the commands:

```
{primary:node0}[edit]
ibm@J58S-1# set security nat source pool pool_name address first_IP_address to
last_IP_address
```

In the command:

- Pool_name is the name of the source address pool we want to create
- First_IP_address is the first IP address in the pool
- Last_IP_address is the last IP address in the pool.

If you do not specify the last IP address, only one address will be added to the pool. You can also use a mask to the first_IP_address to specify multiple IP addresses to the pool.

To configure an optional pool utilization alarm, use the command:

```
{primary:node0}[edit]
ibm@J58S-1# set security nat source pool-utilization-alarm raise-threshold
raise_percentage clear-threshold clear_percentage
```

To configure an optional address persistency, use the command:

```
{primary:node0}[edit]
ibm@J58S-1# set security nat source address-persistent
```

To configure source NAT rules, use the commands:

- For specifying the source of packets for NAT:

```
{primary:node0}[edit]
ibm@J58S-1# set security nat source rule-set rule_set_name from source
```

In the command:

- Rule_set_name is the name of the rule set
- Source is the source for which we need NAT. The source can be zone, interface, or routing-instance level

- For specifying the destination of packets for NAT:

```
{primary:node0}[edit]
ibm@J58S-1# set security nat source rule-set rule_set_name to destination
```

In the command:

- Rule_set_name is the name of the rule set
- Destination is the destination for which we need NAT. The destination can be zone, interface, or routing-instance level

- For defining the matching packets source address that are subject of this NAT rule:

```
{primary:node0}[edit]
ibm@J58S-1# set security nat source rule-set rule_set_name rule rule_name match
source-address IP_address
```

In the command:

- Rule_set_name is the name of the rule set
- Rule_name is the name of the rule
- IP_address is the source IP address of the traffic

- For defining the matching packets destination address that are subject to this NAT rule:

```
{primary:node0}[edit]
ibm@J58S-1# set security nat source rule-set rule_set_name rule rule_name match
destination-address IP_address
```

In the command:

- Rule_set_name is the name of the rule set
- Rule_name is the name of the rule
- IP_address is the destination IP address of the traffic

- For defining the address pool from which we translate the source IP address of the traffic:

```
{primary:node0}[edit]
ibm@J58S-1# set security nat source rule-set rule_set_name rule rule_name then
source-nat pool pool_name
```

In the command:

- Rule_set_name is the name of the rule set
- Rule_name is the name of the rule
- Pool_name is the name of the source address pool we want to use for this rule

- For defining that we translate the source IP address of the traffic with the outgoing interface IP address:

```
{primary:node0}[edit]
ibm@J58S-1# set security nat source rule-set rule_set_name rule rule_name then
source-nat interface
```

In the command:

- Rule_set_name is the name of the rule set
- Rule_name is the name of the rule

To configure NAT proxy if necessary, use the command:

```
{primary:node0}[edit]
ibm@J58S-1# set security nat proxy-arp interface interface address IP_address
```

In the command:

- Interface is the name of the interface in type-fpc/pic/port.logical-unit-number notation
- IP_address is the IP address for which to perform proxy arp

To verify source nat configuration, use the command:

```
{primary:node0}[edit]
ibm@J58S-1# show security nat source
```

To verify source nat operation, use the command:

```
{primary:node0}
ibm@J58S-1> show security nat source rule all
```


To verify source pool operation for nat, use the command:

```
{primary:node0}
ibm@J58S-1> show security nat source pool all
```

To view a summary of source nat operation, use the command:

```
{primary:node0}
ibm@J58S-1> show security nat source summary
```

Disabling port randomization for source NAT

For pool-based source NAT and interface NAT, port numbers are allocated randomly by default. Although randomized port number allocation can provide protection from security threats such as DNS poison attacks, it can also affect performance and memory usage for pool-based source NAT.

To disable port randomization, use the command:

```
{primary:node0}[edit]
ibm@J58S-1# set security nat source port-randomization disable
```

8.5.7 Source NAT configuration example

In Example 8-6, we used the security zone definition from Example 8-2 on page 201 and security policy definition from Example 8-3 on page 212, and based on this preconfiguration we demonstrate three source NAT translations with the following characteristics:

- ▶ Define a source rule-set named **rule-set1** from zone inside to interface ge-1/1/0.0
- ▶ Define a source address pool **spool1** with IP of 10.10.10.11
- ▶ Define a source address pool **spool2** with IP of 10.10.10.15 to 10.10.10.20 and disable port translation for this pool
- ▶ Define a source NAT rule named **rule1** that translates the source IP of all traffic to 10.10.10.100 with the IP of outgoing interface
- ▶ Define a source NAT rule named **rule2** that translates the source IP of all traffic from 192.168.1.101 and 192.168.1.102 to 10.10.10.101 with an IP from source pool **spool1**
- ▶ Define a source NAT rule named **rule3** that translates the source IP of all traffic to 10.10.10.102 and 10.10.10.103 with an IP from source pool **spool2**
- ▶ For proper operation of this setup, configure proxy arp on interface ge-1/1/0.0 for IP addresses from 10.10.10.11 to 10.10.10.20

Example 8-6 Source NAT configuration

```
{primary:node0}[edit]
ibm@J58S-1# edit security nat source

{primary:node0}[edit security nat source]
ibm@J58S-1# set pool spool1 address 10.10.10.11

{primary:node0}[edit security nat source]
ibm@J58S-1# set pool spool2 address 10.10.10.15 to 10.10.10.20

{primary:node0}[edit security nat source]
ibm@J58S-1# set pool spool2 port no-translation

{primary:node0}[edit security nat source]
```

```

ibm@J58S-1# set rule-set rule-set1 from zone inside

{primary:node0}[edit security nat source]
ibm@J58S-1# set rule-set rule-set1 to interface ge-1/1/0.0

{primary:node0}[edit security nat source]
ibm@J58S-1# set rule-set rule-set1 rule rule1 match destination-address
10.10.10.100

{primary:node0}[edit security nat source]
ibm@J58S-1# set rule-set rule-set1 rule rule1 then source-nat interface

{primary:node0}[edit security nat source]
ibm@J58S-1# set rule-set rule-set1 rule rule2 match source-address 192.168.1.101

{primary:node0}[edit security nat source]
ibm@J58S-1# set rule-set rule-set1 rule rule2 match source-address 192.168.1.102

{primary:node0}[edit security nat source]
ibm@J58S-1# set rule-set rule-set1 rule rule2 match destination-address
10.10.10.101

{primary:node0}[edit security nat source]
ibm@J58S-1# set rule-set rule-set1 rule rule2 then source-nat pool spool1

{primary:node0}[edit security nat source]
ibm@J58S-1# set rule-set rule-set1 rule rule3 match destination-address
10.10.10.103

{primary:node0}[edit security nat source]
ibm@J58S-1# set rule-set rule-set1 rule rule3 match destination-address
10.10.10.102

{primary:node0}[edit security nat source]
ibm@J58S-1# set rule-set rule-set1 rule rule3 then source-nat pool spool2

{primary:node0}[edit security nat source]
ibm@J58S-1# top

{primary:node0}[edit]
ibm@J58S-1# set security nat proxy-arp interface ge-1/1/0.0 address 10.10.10.11 to
10.10.10.20

{primary:node0}[edit]
ibm@J58S-1# commit
node0:
configuration check succeeds
node1:
commit complete
node0:
commit complete

{primary:node0}[edit]
ibm@J58S-1# show security nat
source {

```

```

pool spool1 {
    address {
        10.10.10.11/32;
    }
}
pool spool2 {
    address {
        10.10.10.15/32 to 10.10.10.20/32;
    }
    port no-translation;
}
rule-set rule-set1 {
    from zone inside;
    to interface ge-1/1/0.0;
    rule rule1 {
        match {
            destination-address 10.10.10.100/32;
        }
        then {
            source-nat {
                interface;
            }
        }
    }
    rule rule2 {
        match {
            source-address [ 192.168.1.101/32 192.168.1.102/32 ];
            destination-address 10.10.10.101/32;
        }
        then {
            source-nat {
                pool {
                    spool1;
                }
            }
        }
    }
    rule rule3 {
        match {
            destination-address [ 10.10.10.103/32 10.10.10.102/32 ];
        }
        then {
            source-nat {
                pool {
                    spool2;
                }
            }
        }
    }
}
}
proxy-arp {
    interface ge-1/1/0.0 {
        address {
            10.10.10.11/32 to 10.10.10.20/32;
        }
    }
}

```

```

    }
  }
}

```

{primary:node0}[edit]

ibm@J58S-1# **run show security nat source pool all node 0**

node0:

Total pools: 2

```

Pool name      : spool1
Pool id        : 5
Routing instance : default
Host address base : 0.0.0.0
Port           : [1024, 63487]
Total addresses  : 1
Translation hits : 22
Address range    10.10.10.11 - 10.10.10.11    Single Ports  Twin Ports
                                                0              0

```

```

Pool name      : spool2
Pool id        : 6
Routing instance : default
Host address base : 0.0.0.0
Port           : no translation
Total addresses  : 6
Translation hits : 0
Address range    10.10.10.15 - 10.10.10.20    Single Ports  Twin Ports
                                                0              0

```

{primary:node0}[edit]

ibm@J58S-1# **run show security nat source rule all node 0**

node0:

Total rules: 3

```

source NAT rule: rule1          Rule-set: rule-set1
Rule-Id                        : 1
Rule position                   : 1
From zone                      : inside
To interface                   : ge-1/1/0.0
Destination addresses          : 10.10.10.100 - 10.10.10.100
Action                         : interface
Persistent NAT type            : N/A
Inactivity timeout             : 0
Max session number             : 0
Translation hits               : 16

```

```

source NAT rule: rule2          Rule-set: rule-set1
Rule-Id                        : 2
Rule position                   : 2
From zone                      : inside
To interface                   : ge-1/1/0.0
Match
Source addresses               : 192.168.1.101 - 192.168.1.101

```

```

192.168.1.102 - 192.168.1.102
Destination addresses : 10.10.10.101 - 10.10.10.101
Action : spool1
Persistent NAT type : N/A
Inactivity timeout : 0
Max session number : 0
Translation hits : 22

source NAT rule: rule3 Rule-set: rule-set1
Rule-Id : 3
Rule position : 3
From zone : inside
To interface : ge-1/1/0.0
Destination addresses : 10.10.10.103 - 10.10.10.103
10.10.10.102 - 10.10.10.102
Action : spool2
Persistent NAT type : N/A
Inactivity timeout : 0
Max session number : 0
Translation hits : 25

```

{primary:node0}[edit]

ibm@J58S-1# **run show security nat source summary node 0**

node0:

Total pools: 2

Pool	Address	Routing	PAT	Total
Name	Range	Instance		Address
spool1	10.10.10.11-10.10.10.11	default	yes	1
spool2	10.10.10.15-10.10.10.20	default	no	6

Total rules: 3

Rule name	Rule set	From	To	Action
rule1	rule-set1	inside	ge-1/1/0.0	interface
rule2	rule-set1	inside	ge-1/1/0.0	spool1
rule3	rule-set1	inside	ge-1/1/0.0	spool2

{primary:node0}[edit]

ibm@J58S-1#

Persistent NAT

Persistent NAT allows applications to use the Session Traversal Utilities for NAT (STUN) protocol when passing through NAT firewalls. Persistent NAT ensures that all requests from the same internal transport address are mapped to the same reflexive transport address (the public IP address and port created by the NAT device closest to the STUN server).

The following types of persistent NAT can be configured on the IBM j-type Ethernet Appliances:

- Any remote host** All requests from a specific internal IP address and port are mapped to the same reflexive transport address. Any external host can send a packet to the internal host by sending the packet to the reflexive transport address.
- Target host** All requests from a specific internal IP address and port are mapped to the same reflexive transport address. An external host can send a

packet to an internal host by sending the packet to the reflexive transport address. The internal host must have previously sent a packet to the external host's IP address.

Target host port All requests from a specific internal IP address and port are mapped to the same reflexive transport address. An external host can send a packet to an internal host by sending the packet to the reflexive transport address. The internal host must have previously sent a packet to the external host's IP address and port.

You configure any of the persistent NAT types with source NAT rules. The source NAT rule action can use a source NAT pool (with or without port translation) or an egress interface. Persistent NAT is not applicable for destination NAT, because persistent NAT bindings are based on outgoing sessions from internal to external.

Port overloading: Port overloading is used in Junos OS only for normal interface NAT traffic. Persistent NAT does not support port overloading, and you must explicitly disable port overloading with the port-overloading off option at the [edit security nat source] hierarchy level.

To configure security policies to permit or deny persistent NAT traffic, you can use two new predefined services: *junos-stun* and *junos-persistent-nat*.

Persistent NAT versus persistent address feature: Persistent NAT is different from the persistent address feature. The persistent address feature applies to address mappings for source NAT pools configured on the device. The persistent NAT feature applies to address mappings on an external NAT device, and is configured for a specific source NAT pool or egress interface. Also, persistent NAT is intended for use with STUN client/server applications.

Session Traversal Utilities for NAT (STUN) Protocol

Many video and voice applications do not work properly in a NAT environment. For example, Session Initiation Protocol (SIP), used with VoIP, encodes IP addresses and port numbers within application data. If a NAT firewall exists between the requestor and receiver, the translation of the IP address and port number in the data invalidates the information.

Also, a NAT firewall does not maintain a pinhole for incoming SIP messages. This forces the SIP application to either constantly refresh the pinhole with SIP messages or use an ALG to track registration, a function that might or might not be supported by the gateway device.

The Session Traversal Utilities for NAT (STUN) protocol, first defined in RFC 3489, Simple Traversal of User Datagram Protocol (UDP) through Network Address Translators (NATs) and then later in RFC 5389, Session Traversal Utilities for NAT, is a simple client/server protocol. A STUN client sends requests to a STUN server, which returns responses to the client. A STUN client is usually part of an application that requires a public IP address and port. STUN clients can reside in an end system such as a PC or in a network server whereas STUN servers are usually attached to the public Internet.

Note: Both the STUN client and STUN server must be provided by the application. Juniper Networks does not provide a STUN client or server.

The STUN protocol allows a client to:

- Discover whether the application is behind a NAT firewall.

- Determine the type of NAT binding being used.
- Learn the reflexive transport address, which is the IP address and port binding allocated by NAT device closest to the STUN server. (There might be multiple levels of NAT between the STUN client and the STUN server.)

The client application can use the IP address binding information within protocols such as SIP and H.323.

Persistent NAT Configuration Overview

To configure persistent NAT, specify the following options with the source NAT rule action (for either a source NAT pool or an egress interface):

- The type of persistent NAT: One of the following: any remote host, target host, or target host port.
- (Optional) Address mapping: This option allows requests from a specific internal IP address to be mapped to the same reflexive IP address; internal and reflexive ports can be any ports. An external host using any port can send a packet to the internal host by sending the packet to the reflexive IP address (with a configured incoming policy that allows external to internal traffic). If this option is not configured, the persistent NAT binding is for specific internal and reflexive transport addresses.

You can only specify the address-mapping option when the persistent NAT type is any remote host and the source NAT rule action is one of the following actions:

- Source NAT pool with IP address shifting
- Source NAT pool with no port translation and no overflow pool
- (Optional) Inactivity timeout: Time, in seconds, that the persistent NAT binding remains in the device's memory when all the sessions of the binding entry have expired. When the configured timeout is reached, the binding is removed from memory. The default value is 300 seconds. Configure a value from 60 through 7200 seconds.

When all sessions of a persistent NAT binding have expired, the binding remains in a query state in the IBM j-type Ethernet Appliance's memory for the specified inactivity timeout period. The query binding is automatically removed from memory when the inactivity timeout period expires (the default is 300 seconds). You can explicitly remove all or specific persistent NAT query bindings with the `clear security nat source persistent-nat-table` command.

- (Optional) Maximum session number: Maximum number of sessions with which a persistent NAT binding can be associated. The default is 30 sessions. Configure a value from 8 through 100.

For interface NAT, you need to explicitly disable port overloading with the *port-overloading off* option at the *[edit security nat source]* hierarchy level.

Finally, there are two predefined services that you can use in security policies to permit or deny STUN and persistent NAT traffic:

- *Junos-stun*: STUN protocol traffic
- *Junos-persistent-nat*: Persistent NAT traffic

For the any remote host persistent NAT type, the direction of the security policy is from external to internal. For target host or target host port persistent NAT types, the direction of the security policy is from internal to external.

For more information about the persistent NAT, refer to the *Junos OS Security Configuration Guide - Release 10.3* available at:

<http://www.juniper.net/techpubs/software/junos-srx/junos-srx10.3/index.html>

8.6 Virtual Private Networks

This section presents a brief overview of VPNs and how to configure and monitor them on IBM j-type security appliances.

VPN overview

A VPN provides a means for securely communicating among remote computers across a public WAN such as the Internet.

A VPN connection can link two LANs (site-to-site VPN) or a remote dial-up user and a LAN. The traffic that flows between these two points passes through shared resources such as routers, switches, and other network equipment that make up the public WAN. To secure VPN communication while passing through the WAN, the two participants create an Internet Protocol Security (IPsec) tunnel.

8.6.1 Internet Protocol Security

Internet Protocol Security (IPsec) is a suite of related protocols for cryptographically securing communications at the IP Packet Layer. IPsec also provides methods for the manual and automatic negotiation of security associations (SAs) and key distribution, all the attributes for which are gathered in a Domain of Interpretation (DOI). The IPsec DOI is a document containing definitions for all the security parameters required for the successful negotiation of a VPN tunnel—essentially, all the attributes required for SA and IKE negotiations. See RFC 2407 and RFC 2408 for more information.

Security associations

A security association (SA) is an unidirectional agreement between the VPN participants regarding the methods and parameters to use in securing a communication channel. Full bidirectional communication requires at least two SAs, one for each direction.

Through the SA, an IPsec tunnel can provide the following security functions:

- ▶ Privacy (through encryption)
- ▶ Content integrity (through data authentication)
- ▶ Sender authentication and—if using certificates—nonrepudiation (through data origin authentication)

The security functions you employ depend on your needs. If you need only to authenticate the IP packet source and content integrity, you can authenticate the packet without applying any encryption. On the other hand, if you are concerned only with preserving privacy, you can encrypt the packet without applying any authentication mechanisms. Optionally, you can both encrypt and authenticate the packet. Most network security designers choose to encrypt, authenticate, and replay-protect their VPN traffic.

An IPsec tunnel consists of a pair of unidirectional SAs—one at each end of the tunnel—that specify the security parameter index (SPI), destination IP address, and security protocol (Authentication Header or Encapsulating Security Payload) employed.

An SA groups together the following components for securing communications:

- ▶ Security algorithms and keys
- ▶ Protocol mode, either transport or tunnel. Junos devices always use tunnel mode
- ▶ Key-management method, either manual key or AutoKey IKE.
- ▶ SA lifetime

Note: Junos does not support IPsec in transport mode or L2TP over IPsec.

For inbound traffic, Junos software looks up the SA by using the following triplet:

- ▶ Destination IP address.
- ▶ Security protocol, either AH or ESP.
- ▶ Security parameter index (SPI) value.

For outbound VPN traffic, the policy invokes the SA associated with the VPN tunnel.

IPsec key management

The distribution and management of keys are critical to using VPNs successfully. Junos software supports IPsec technology for creating VPN tunnels with three kinds of key creation mechanisms:

- ▶ Manual key
- ▶ AutoKey IKE with a preshared key or a certificate
- ▶ Diffie-Hellman (DH) key

You can choose your key creation mechanism—also called authentication method—during Phase 1 and Phase 2 proposal configuration.

Manual key

With manual keys, administrators at both ends of a tunnel configure all the security parameters. This is a viable technique for small, static networks where the distribution, maintenance, and tracking of keys are not difficult. However, safely distributing manual-key configurations across great distances poses security issues. Aside from passing the keys face-to-face, you cannot be completely sure that the keys have not been compromised while in transit. Also, whenever you want to change the key, you are faced with the same security issues as when you initially distributed it.

AutoKey IKE

When you need to create and manage numerous tunnels, you need a method that does not require you to configure every element manually. IPsec supports the automated generation and negotiation of keys and security associations using the Internet Key Exchange (IKE) protocol. Junos software refers to such automated tunnel negotiation as AutoKey IKE and supports AutoKey IKE with preshared keys and AutoKey IKE with certificates:

- ▶ AutoKey IKE with preshared keys: Using AutoKey IKE with preshared keys to authenticate the participants in an IKE session, each side must configure and securely exchange the preshared key in advance. In this regard, the issue of secure key distribution is the same as that with manual keys. However, after distributed, an autokey, unlike a manual key, can automatically change its keys at predetermined intervals using the IKE protocol. Frequently changing keys greatly improves security, and automatically doing so greatly reduces key-management responsibilities. However, changing keys increases traffic overhead; therefore, doing so too often can reduce data transmission efficiency. A preshared key is a key for both encryption and decryption, which both participants must have before initiating communication.
- ▶ AutoKey IKE with certificates: When using certificates to authenticate the participants during an AutoKey IKE negotiation, each side generates a public-private key pair and acquires a certificate. As long as the issuing certificate authority (CA) is trusted by both sides, the participants can retrieve the peer's public key and verify the peer's signature. There is no need to keep track of the keys and SAs because IKE does it automatically.

Diffie-Hellman exchange

A Diffie-Hellman (DH) exchange allows the participants to produce a shared secret value. The strength of the technique is that it allows the participants to create the secret value over an unsecured medium without passing the secret value through the wire. There are five DH groups, and Junos software supports groups 1, 2, and 5.

The size of the prime modulus used in each group's calculation differs as follows:

- ▶ DH Group 1—768-bit modulus
- ▶ DH Group 2—1024-bit modulus
- ▶ DH Group 5—1536-bit modulus

DH Group 1: The strength of DH Group 1 security depreciated; therefore, we do not recommend its use.

The larger the modulus, the more secure the generated key is considered to be; however, the larger the modulus, the longer the key-generation process takes. Because the modulus for each DH group is a different size, the participants must agree to use the same group.

Note: If you configure multiple (up to four) proposals for Phase 1 negotiations, use the same DH group in all proposals. The same guideline applies to multiple proposals for Phase 2 negotiations.

IPsec Security Protocols

IPsec uses two protocols to secure communications at the IP layer:

- ▶ Authentication Header (AH): A security protocol for authenticating the source of an IP packet and verifying the integrity of its content
- ▶ Encapsulating Security Payload (ESP): A security protocol for encrypting the entire IP packet (and authenticating its content)

You can choose your security protocols—also called authentication and encryption algorithms—during Phase 1 and Phase 2 proposal configuration.

AH Protocol

The Authentication Header protocol provides a means to verify the authenticity/integrity of the content and origin of a packet. You can authenticate the packet by the checksum calculated through a Hash Message Authentication Code (HMAC) using a secret key and either MD5 or SHA-1 hash functions:

- ▶ Message Digest 5 (MD5): An algorithm that produces a 128-bit hash (also called a digital signature or message digest) from a message of arbitrary length and a 16-byte key. The resulting hash is used, such as a fingerprint of the input, to verify content and source authenticity and integrity.
- ▶ Secure Hash Algorithm (SHA-1): An algorithm that produces a 160-bit hash from a message of arbitrary length and a 20-byte key. It is generally regarded as more secure than MD5 because of the larger hashes it produces. Because the computational processing is done in the ASIC, the performance cost is negligible.

Note: For more information about MD5 hashing algorithms, see RFC 1321 and RFC 2403. For more information about SHA hashing algorithms, see RFC 2404. For more information about HMAC, see RFC 2104.

ESP protocol

The Encapsulating Security Payload (ESP) protocol provides a means to ensure privacy (encryption) and source authentication and content integrity (authentication). ESP in tunnel mode encapsulates the entire IP packet (header and payload) and then appends a new IP header to the now-encrypted packet. This new IP header contains the destination address needed to route the protected data through the network.

With ESP, you can both encrypt and authenticate, encrypt only, or authenticate only.

For encryption, you can choose one of the following encryption algorithms:

- ▶ Data Encryption Standard (DES): A cryptographic block algorithm with a 56-bit key.
- ▶ Triple DES (3DES): A more powerful version of DES in which the original DES algorithm is applied in three rounds, using a 168-bit key. DES provides a significant performance savings but is considered unacceptable for many classified or sensitive material transfers.
- ▶ Advanced Encryption Standard (AES): An emerging encryption standard which, when adopted by Internet infrastructures worldwide, will offer greater interoperability with other devices. Junos software supports AES with 128-bit, 192-bit, and 256-bit keys.

For authentication, you can use either the MD5 or the SHA-1 algorithm.

Encryption: Even though it is possible to select NULL for encryption, it has been demonstrated that IPsec might be vulnerable to attack under such circumstances. Therefore, we suggest that you choose an encryption algorithm for maximum security.

IPsec tunnel negotiation

To establish an AutoKey IKE IPsec tunnel, two phases of negotiation are required:

- ▶ In Phase 1, the participants establish a secure channel in which to negotiate the IPsec security associations (SAs).
- ▶ In Phase 2, the participants negotiate the IPsec SAs for encrypting and authenticating the ensuing exchanges of user data.

For a manual key IPsec tunnel, because all of the SA parameters have been previously defined, there is no need to negotiate which SAs to use. In essence, the tunnel has already been established. When traffic matches a policy using that manual key tunnel or when a route involves the tunnel, the IBM j-type security appliance simply encrypts and authenticates the data, as you determined, and forwards it to the destination gateway.

Phase 1 of IKE tunnel negotiation

Phase 1 of an AutoKey IKE tunnel negotiation consists of the exchange of proposals for how to authenticate and secure the channel. The participants exchange proposals for acceptable security services such as:

- ▶ Encryption algorithms (DES and 3DES) and authentication algorithms (MD5 and SHA-1). (For more information, see “IPsec Security Protocols” on page 246.)
- ▶ A Diffie-Hellman group. (For more information, see “Diffie-Hellman exchange” on page 246.)
- ▶ Preshared Key or RSA/DSA certificates. (For more information, see “IPsec key management” on page 245.)

A successful Phase 1 negotiation concludes when both ends of the tunnel agree to accept at least one set of the Phase 1 security parameters proposed and then process them. IBM j-type

Networks devices support up to four proposals for Phase 1 negotiations, allowing you to define how restrictive a range of security parameters for key negotiation you will accept.

The predefined Phase 1 proposals that Junos software provides are:

- ▶ Standard: Pre-g2-aes128-sha and pre-g2-3des-sha
- ▶ Compatible: Pre-g2-3des-sha, pre-g2-3des-md5, pre-g2-des-sha, and pre-g2-des-md5
- ▶ Basic: Pre-g1-des-sha and pre-g1-des-md5

You can also define custom Phase 1 proposals.

Note: If you are using the dynamic VPN feature, note that you must create a custom Phase 1 proposal. Predefined Phase 1 proposals are not available at this time.

Phase 1 exchanges can take place in either main or aggressive mode. You can choose your mode during IKE policy configuration:

- ▶ Main Mode: In main mode, the initiator and recipient send three two-way exchanges (six messages total) to accomplish the following services:
 - First exchange (messages 1 and 2): Propose and accept the encryption and authentication algorithms.
 - Second exchange (messages 3 and 4): Execute a Diffie-Hellman exchange, and the initiator and recipient each provide a pseudo-random number.
 - Third exchange (messages 5 and 6): Send and verify their identities.

The information transmitted in the third exchange of messages is protected by the encryption algorithm established in the first two exchanges. Thus, the participants' identities are not transmitted in the clear.

- ▶ Aggressive Mode: In aggressive mode, the initiator and recipient accomplish the same objectives, but in only two exchanges, with a total of three messages:
 - First message: The initiator proposes the SA, initiates a Diffie-Hellman exchange, and sends a pseudo-random number and its IKE identity.
 - Second message: The recipient accepts the SA; authenticates the initiator; and sends a pseudo-random number, its IKE identity, and, if using certificates, the recipient's certificate.
 - Third message: The initiator authenticates the recipient, confirms the exchange, and, if using certificates, sends the initiator's certificate.

Because the participants' identities are exchanged in the clear (in the first two messages), aggressive mode does not provide identity protection.

Note: When a dialup VPN user negotiates an AutoKey IKE tunnel with a preshared key, aggressive mode must be used. Therefore, you must always use aggressive mode with the dynamic VPN feature. Note also that a dialup VPN user can use an email address, a fully qualified domain name (FQDN), or an IP address as its IKE ID. A dynamic peer can use either an email address or FQDN, but not an IP address.

Phase 2 of IKE tunnel negotiation

After the participants establish a secure and authenticated channel, they proceed through Phase 2, in which they negotiate the SAs to secure the data to be transmitted through the IPsec tunnel.

Similar to the process for Phase 1, the participants exchange proposals to determine which security parameters to employ in the SA. A Phase 2 proposal also includes a security protocol—either Encapsulating Security Payload or Authentication Header—and selected encryption and authentication algorithms. The proposal can also specify a Diffie-Hellman group, if Perfect Forward Secrecy (PFS) is desired.

Regardless of the mode used in Phase 1, Phase 2 always operates in quick mode and involves the exchange of three messages.

IBM j-type Networks devices support up to four proposals for Phase 2 negotiations, allowing you to define how restrictive a range of tunnel parameters you will accept. The predefined Phase 2 proposals that Junos software provides are as follows:

- ▶ Standard: g2-esp-3des-sha and g2-esp-aes128-sha
- ▶ Compatible: nopfs-esp-3des-sha, nopfs-esp-3des-md5, nopfs-esp-des-sha, and nopfs-esp-des-md5
- ▶ Basic: nopfs-esp-des-sha and nopfs-esp-des-md5

If you are using the dynamic VPN feature, note that you must create a custom Phase 2 proposal. Predefined Phase 2 proposals are not available at this time.

You can also define custom Phase 2 proposals:

- ▶ Proxy IDs: In Phase 2, the peers exchange proxy IDs. A proxy ID is a three-part tuple consisting of local IP address-remote IP address-service. The proxy ID for both peers must match, which means that the service specified in the proxy ID for both peers must be the same, and the local IP address specified for one peer must be the same as the remote IP address specified for the other peer.

You can specify the format of Proxy IDs during IPsec Autokey configuration.

- ▶ Perfect Forward Secrecy: Perfect Forward Secrecy (PFS) is a method for deriving Phase 2 keys independent from and unrelated to the preceding keys. Alternatively, the Phase 1 proposal creates the key (the SKEYID_d key) from which all Phase 2 keys are derived. The SKEYID_d key can generate Phase 2 keys with a minimum of CPU processing. Unfortunately, if an unauthorized party gains access to the SKEYID_d key, all your encryption keys are compromised.

PFS addresses this security risk by forcing a new Diffie-Hellman key exchange to occur for each Phase 2 tunnel. Using PFS is thus more secure, although the rekeying procedure in Phase 2 might take slightly longer with PFS enabled. You can enable PFS during IPsec policy configuration.

- ▶ Replay Protection: A replay attack occurs when somebody intercepts a series of packets and uses them later either to flood the system, causing a denial-of-service (DoS), or to gain entry to the trusted network. Junos software provides a replay protection feature, which enables devices to check every IPsec packet to see if it has been received previously. If packets arrive outside a specified sequence range, Junos software rejects them. Use of this feature does not require negotiation because packets are always sent with sequence numbers. You simply have the option of checking or not checking the sequence numbers. You can enable replay protection during IPsec Autokey configuration.

Distributed VPNs

In IBM j-type security appliances, the IKE provides tunnel management for IPsec and authenticates end entities. The IKE performs a Diffie-Hellman key exchange to generate an IPsec tunnel between network devices. The IPsec tunnels generated by IKE are used to encrypt, decrypt, and authenticate user traffic between the network devices at the IP layer.

The VPN is created by distributing the IKE and IPsec workload among the multiple Security Processing Units (SPUs) of the platform. The IKE workload is distributed based on a key generated from the IKE packet's 4 tuples (source IP address, destination IP addresses, and UDP ports). The workload is distributed by assigning anchoring SPUs logically and mapping the logical SPUs to physical SPU-based on the composition at that given time. This distribution prevents any change in the number and composition of SPUs in the device, which might happen due to hot swap or SPC failure. The SPU in a device communicates with the Routing Engine to create a distributed VPN.

In IPsec, the workload is distributed by the same algorithm that distributes the IKE. The Phase 2 SA for a given VPN tunnel termination points pair is exclusively owned by a particular SPU, and all IPsec packets belonging to this Phase 2 SA are forwarded to the anchoring SPU of that security association for IPsec processing.

Understanding IKE and IPsec packet processing

An IPsec VPN tunnel consists of tunnel setup and applied security. During tunnel setup, the peers establish security associations (SAs), which define the parameters for securing traffic between themselves. After the tunnel is established, IPsec protects the traffic sent between the two tunnel endpoints by applying the security parameters defined by the SAs during tunnel setup. Within the Junos implementation, IPsec is applied in tunnel mode, which supports the Encapsulating Security Payload (ESP) and Authentication Header (AH) protocols.

Packet processing in tunnel mode

IPsec operates in one of two modes—transport or tunnel. When both ends of the tunnel are hosts, you can use either mode. When at least one of the endpoints of a tunnel is a security gateway, such as a Junos router or firewall, you must use tunnel mode. IBM j-type security appliances always operate in tunnel mode for IPsec tunnels.

In tunnel mode, the entire original IP packet (payload and header) is encapsulated within another IP payload and a new header is appended to it. The entire original packet can be encrypted, authenticated, or both. With the Authentication Header (AH) protocol, the AH and new headers are also authenticated. With the Encapsulating Security Payload (ESP) protocol, the ESP header can also be authenticated.

In a site-to-site VPN, the source and destination addresses used in the new header are the IP addresses of the outgoing interface.

In a dial-up VPN, there is no tunnel gateway on the VPN dial-up client end of the tunnel; the tunnel extends directly to the client itself. In this case, on packets sent from the dial-up client, both the new header and the encapsulated original header have the same IP address: that of the client's computer.

Note: Junos does not support IPsec in transport mode

IKE packet processing

When a cleartext packet arrives on an IBM j-type security appliance that requires tunneling and no active Phase 2 SA exists for that tunnel, Junos software begins IKE negotiations and drops the packet. The source and destination addresses in the IP packet header are those of the local and remote IKE gateways, respectively. In the IP packet payload, there is a UDP segment encapsulating an ISAKMP (IKE) packet. The format for IKE packets is the same for Phase 1 and Phase 2.

Meanwhile, the source host has sent the dropped packet again. Typically, by the time the second packet arrives, IKE negotiations are complete and Junos software protects it—and all subsequent packets in the session—with IPsec before forwarding it.

Note: Junos does not support yet IKE termination on a virtual router instance (VR). This feature is on the roadmap in a future release of Junos software.

IPsec packet processing

After IKE negotiations complete and the two IKE gateways have established Phase 1 and Phase 2 security associations, the device applies IPsec protection to subsequent cleartext IP packets that hosts behind one IKE gateway send to hosts behind the other gateway (assuming that policies permit the traffic). If the Phase 2 SA specifies the Encapsulating Security Protocol in tunnel mode, the packet looks similar to Figure 8-2. The device adds two additional headers to the original packet that the initiating host sends.

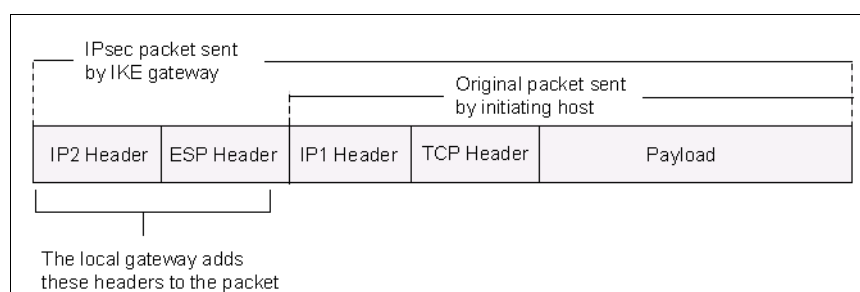


Figure 8-2 IPsec Packet: ESP in Tunnel Mode

8.6.2 IPsec VPN configuration overview

IKE IPsec tunnel negotiation occurs in two phases. In Phase 1, participants establish a secure channel in which to negotiate the IPsec security association (SA). In Phase 2, participants negotiate the IPsec SA for authenticating traffic that will flow through the tunnel. Just as there are two phases to tunnel negotiation, there are two phases to tunnel configuration.

The following procedure lists the recommended order in which you must configure an IPsec VPN tunnel:

1. Configure Phase 1 of the IPsec tunnel:
 - a. Configure an IKE Phase 1 proposal.
 - b. Configure an IKE policy that references the proposal.
 - c. Configure an IKE gateway that references the policy.
2. Configure Phase 2 of the IPsec tunnel:
 - a. Configure a Phase 2 proposal.
 - b. Configure a policy that references the proposal.
 - c. Configure an Autokey IKE that references the policy and the gateway.
3. Update your global VPN settings.

To configure an IKE Phase 1 proposal, use the following commands:

- For selecting the ike proposal authentication method:

```
{primary:node0}[edit]
ibm@J58S-1# set security ike proposal proposal_name authentication-method
authentication_method
```

In the command:

- Proposal_name is the name of the ike proposal we create
- Authentication_method is one of the pre-shared-keys, or rsa-signatures authentication methods options

- For selecting the ike proposal Diffie-Hellman group type:

```
{primary:node0}[edit]
ibm@J58S-1# set security ike proposal proposal_name dh-group dh_group
```

In the command:

- Proposal_name is the name of the ike proposal we create
- dh_group is one of the group1, group2, or group5 Diffie-Hellman group options

- For selecting the ike proposal authentication algorithm:

```
{primary:node0}[edit]
ibm@J58S-1# set security ike proposal proposal_name authentication-algorithm
authentication_algorithm
```

In the command:

- Proposal_name is the name of the ike proposal we create
- Authentication_algorithm is one of the sha1, sha-256, or md5 algorithms options

- For selecting the ike proposal encryption algorithm:

```
{primary:node0}[edit]
ibm@J58S-1# set security ike proposal proposal_name encryption-algorithm
encryption_algorithm
```

In the command:

- Proposal_name is the name of the IKE proposal we create
- Encryption_algorithm is one of the des-cbc, 3des-cbc, aes-128-cbc, aes-192-cbc or aes-256-cbg encryption algorithms options

- For selecting the ike proposal lifetime in seconds:

```
{primary:node0}[edit]
ibm@J58S-1# set security ike proposal proposal_name lifetime-seconds time
```

In the command:

- Proposal_name is the name of the IKE proposal we create
- Time is the lifetime of the ike in seconds

- For adding a description to the ike proposal:

```
{primary:node0}[edit]
ibm@J58S-1# set security ike proposal proposal_name description "text"
```

In the command:

- Proposal_name is the name of the IKE proposal we create
- Text is the description of the ike proposal

To configure an IKE policy that references the IKE proposal, use these commands:

- For selecting the ike policy mode:

```
{primary:node0}[edit]
ibm@J58S-1# set security ike policy policy_name mode mode
```


In the command:

- Policy_name is the name of the IKE policy we create
- Mode is one of the main or aggressive mode options.

- For referencing an ike proposal in the IKE policy:

```
{primary:node0}[edit]  
ibm@J58S-1# set security ike policy policy_name proposals proposal
```

In the command:

- Policy_name is the name of the IKE policy we create
- Proposal is the IKE proposal we need to reference

- For configuring a preshared key in the IKE policy:

```
{primary:node0}[edit]  
ibm@J58S-1# set security ike policy policy_name pre-shared-key ascii-text key
```

In the command:

- Policy_name is the name of the IKE policy we create
- Key is the preshared key we use for IKE authentication

- For adding a description to the ike policy:

```
{primary:node0}[edit]  
ibm@J58S-1# set security ike policy policy_name description "text"
```

In the command:

- Policy_name is the name of the IKE policy we create
- Text is the description of the IKE policy

To configure an IKE gateway that references the IKE policy, use the commands:

- For referencing an IKE policy in the IKE gateway:

```
{primary:node0}[edit]  
ibm@J58S-1# set security ike gateway gateway_name ike-policy policy_name
```

In the command:

- Gateway_name is the name of the IKE gateway we create
- Policy_name is the name of the IKE policy we need to reference

- For specifying an IKE gateway IP address:

```
{primary:node0}[edit]  
ibm@J58S-1# set security ike gateway gateway_name address IP_address
```

In the command:

- Gateway_name is the name of the IKE gateway we create
- IP_address is the IP address of the remote IKE gateway

- For specifying an IKE gateway external interface:

```
{primary:node0}[edit]  
ibm@J58S-1# set security ike gateway gateway_name external-interface interface_name
```

In the command:

- Gateway_name is the name of the IKE gateway we create
- Interface_name is the external interface where we enable the IKE protocol

- For specifying an IKE gateway dead peer detection interval:

```
{primary:node0}[edit]
```

```
ibm@J58S-1# set security ike gateway gateway_name dead-peer-detection interval interval
```

In the command:

- Gateway_name is the name of the IKE gateway we create
- Interval is the dead peer detection interval between 10 and 60 seconds

- For specifying an IKE gateway dead peer detection threshold:

```
{primary:node0}[edit]
```

```
ibm@J58S-1# set security ike gateway gateway_name dead-peer-detection threshold threshold
```

In the command:

- Gateway_name is the name of the IKE gateway we create
- Threshold is the dead peer detection threshold between 1 and 5 retransmissions

To configure a Phase 2 ipsec proposal, use the commands:

- For selecting the Phase 2 ipsec proposal authentication algorithm:

```
{primary:node0}[edit]
```

```
ibm@J58S-1# set security ipsec proposal proposal_name authentication-algorithm authentication_algorithm
```

In the command:

- Proposal_name is the name of the ipsec proposal we create
- Authentication_algorithm is one of the hmac-sha1-96 or hmac-md5-96 algorithms options

- For selecting the Phase 2 ipsec proposal encryption algorithm:

```
{primary:node0}[edit]
```

```
ibm@J58S-1# set security ipsec proposal proposal_name encryption-algorithm encryption_algorithm
```

In the command:

- Proposal_name is the name of the ipsec proposal we create
- Encryption_algorithm is one of the des-cbc, 3des-cbc, aes-128-cbc, aes-192-cbc or aes-256-cbg encryption algorithms options

- For selecting the Phase 2 ipsec proposal protocol:

```
{primary:node0}[edit]
```

```
ibm@J58S-1# set security ipsec proposal proposal_name protocol protocol
```

In the command:

- Proposal_name is the name of the ipsec proposal we create
- Protocol is one of the esp or ah protocol options

- For selecting the Phase 2 ipsec proposal lifetime in seconds:

```
{primary:node0}[edit]
```

```
ibm@J58S-1# set security ipsec proposal proposal_name lifetime-seconds time
```

In the command:

- Proposal_name is the name of the ipsec proposal we create
- Time is the lifetime of the ipsec proposal in seconds

- For selecting the Phase 2 ipsec proposal lifetime in kilobytes:

```
{primary:node0}[edit]
```

```
ibm@J58S-1# set security ipsec proposal proposal_name lifetime-kilobytes kB
```

In the command:

- Proposal_name is the name of the ipsec proposal we create
- kB is the lifetime of the ipsec proposal in kilobytes

- For adding a description to the Phase 2 ipsec proposal:

```
{primary:node0}[edit]  
ibm@J58S-1# set security ipsec proposal proposal_name description "text"
```

In the command:

- Proposal_name is the name of the ipsec proposal we create
- Text is the description of the ipsec proposal

To configure an ipsec policy that references the ipsec proposal, use the commands:

- For referencing an ipsec proposal in the ipsec policy:

```
{primary:node0}[edit]  
ibm@J58S-1# set security ipsec policy policy_name proposals proposal
```

In the command:

- Policy_name is the name of the ipsec policy we create
- Proposal is the ipsec proposal we need to reference

- For selecting the ipsec policy perfect forward secrecy Diffie-Hellman group type:

```
{primary:node0}[edit]  
ibm@J58S-1# set security ipsec policy policy_name perfect-forward-secrecy keys  
dh_group
```

In the command:

- Proposal_name is the name of the ipsec policy we create
- dh_group is one of the group1, group2, or group5 Diffie-Hellman group options

- For adding a description to the ipsec policy:

```
{primary:node0}[edit]  
ibm@J58S-1# set security ipsec policy policy_name description "text"
```

In the command:

- Policy_name is the name of the ipsec policy we create
- Text is the description of the ipsec policy

To configure an Autokey IKE that references the policy and the gateway, use the commands:

- For configuring a routed mode VPN tunnel and binding it to the special ipsec interface:

```
{primary:node0}[edit]  
ibm@J58S-1# set security ipsec vpn vpn_name bind-interface interface
```

In the command:

- Vpn_name is the name of the vpn tunnel we create
- Interface is the binding interface for this tunnel (usually is interface st0.0). This configuration is applicable only for routed mode vpn

- For selecting an IKE gateway for the VPN tunnel:

```
{primary:node0}[edit]  
ibm@J58S-1# set security ipsec vpn vpn_name ike gateway gateway_name
```

In the command:

- Vpn_name is the name of the vpn tunnel we create
- Gateway_name is the name of the IKE gateway for this vpn

- For selecting an ipsec policy for the VPN tunnel:

```
{primary:node0}[edit]
ibm@J58S-1# set security ipsec vpn vpn_name ike ipsec-policy policy_name
```

In the command:

- Vpn_name is the name of the vpn tunnel we create
- Policy_name is the name of the ipsec policy for this vpn

To set optional global VPN settings, use the following commands:

- To configure the device to detect and respond to a bad IPsec security parameter index (SPI) before deleting the SA and initiating a new one, use the command:

```
{primary:node0}[edit]
ibm@J58S-1# set security ike respond-bad-spi respond_value
```

In the command, respond_value is the number of times it will detect and respond to a bad IPsec security parameter index (SPI) before deleting the SA and initiating a new one

- To configure the device to monitor the VPN by sending Internet Control Message Protocol (ICMP) requests to the peer, use the command:

```
{primary:node0}[edit]
ibm@J58S-1# set security ipsec vpn-monitor-options interval interval threshold
threshold
```

In the command:

- Interval is the interval between the ICMP requests
- Threshold is the number of unsuccessful ICMP replies before we declare the peer dead

To display the ike configuration, use the command

```
{primary:node0}[edit]
ibm@J58S-1# show security ike
```

To display the ipsec configuration, use the command

```
{primary:node0}[edit]
ibm@J58S-1# show security ipsec
```

To verify the ike operation, use the command:

```
{primary:node0}[edit]
ibm@J58S-1> show security ike security-associations
```

To verify the ipsec operation, use the command:

```
{primary:node0}[edit]
ibm@J58S-1> show security ipsec security-associations
```

To verify the ipsec statistics, use the command:

```
{primary:node0}[edit]
ibm@J58S-1> show security ipsec security-associations
```

8.6.3 VPN configuration example

For IPsec VPN configuration we need two appliances and for this reason we split the J58S chassis cluster, and we are using them individually as presented in Figure 8-3 on page 257.

For simplicity we are presenting the configuration for only one appliance, as the configuration for the second appliance is similar.

The only difference is the IP address for the ge-1/2/0.0 and st0.0 interfaces as presented in the Figure 8-3.

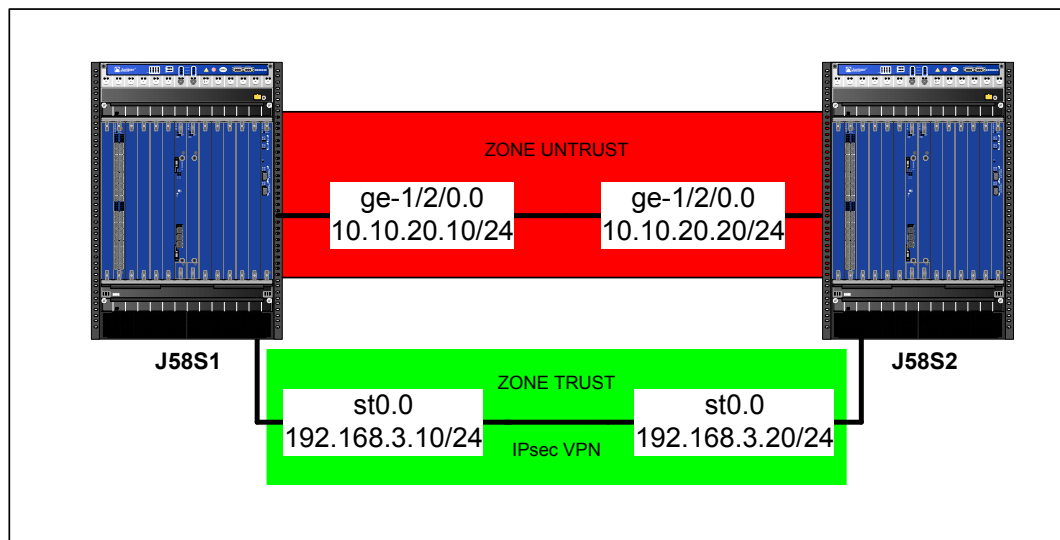


Figure 8-3 Topology used for IPsec configuration example

Example 8-7 on page 258 presents configuration for a routed IPsec tunnel with the following characteristics:

- ▶ Preliminary zone configuration:
 - Configure the interface ge-1/2/0.0 with IP address 10.10.20.10/24
 - Configure the interface st0.0 with IP address 192.168.3.10/24
 - Create a security zone named untrust and add the interface ge-1/2/0 to it
 - Create a security zone named trust and add the interface st0.0 to it
 - Permit all host inbound traffic (protocols and system services) in both zones
 - Permit all traffic between the zones
- ▶ IKE proposal attributes:
 - Proposal name **ike-prop1**
 - Authentication method preshared-key
 - Authentication algorithm **sha-256**
 - Authentication dh-group is **group5**
 - Encryption algorithm **aes-256-cbc**
- ▶ IKE policy attributes:
 - Policy name **ike-pol1**
 - Preshared key is **vpn_key**
 - Negotiation mode is **main**
 - Use the IKE proposal **ike-prop1**
- ▶ IKE gateway attributes:
 - Gateway name **gw1**
 - Gateway IP address 10.10.20.20
 - External interface is ge-1/2/0.0
 - Use the IKE policy **ike-pol1**
 - Dead peer detection interval is 10 seconds

- ▶ IPsec proposal attributes:
 - Proposal name **ipsec-prop1**
 - Encapsulation protocol **esp**
 - Authentication algorithm **hmac-sha1-96**
 - Encryption algorithm **aes-256-cbc**
- ▶ IPsec policy attributes:
 - Policy name **ipsec-pol1**
 - Use IPsec proposal **ipsec-prop1**
- ▶ VPN tunnel attributes:
 - Vpn name is **vpn1**
 - Use IKE gateway **gw1**
 - Use IPsec policy **ipsec-prop1**
 - Bind to interface **st0.0**
 - Create the tunnel immediately

Example 8-7 Routed VPN configuration

```
{primary:node0}[edit]
ibm@J58S-1# set interfaces st0.0 family inet address 192.168.3.10/24

{primary:node0}[edit]
ibm@J58S-1# set security zones security-zone untrust interfaces ge-1/2/0.0

{primary:node0}[edit]
ibm@J58S-1# set security zones security-zone untrust host-inbound-traffic
system-services all

{primary:node0}[edit]
ibm@J58S-1# set security zones security-zone untrust host-inbound-traffic
protocols all

{primary:node0}[edit]
ibm@J58S-1# set security zones security-zone trust interfaces st0.0

{primary:node0}[edit]
ibm@J58S-1# set security zones security-zone trust host-inbound-traffic
system-services all

{primary:node0}[edit]
ibm@J58S-1# set security zones security-zone trust host-inbound-traffic protocols
all

{primary:node0}[edit]
ibm@J58S-1# edit security policies from-zone trust to-zone untrust

{primary:node0}[edit security policies from-zone trust to-zone untrust]
ibm@J58S-1# set policy in-out-1 match source-address any

{primary:node0}[edit security policies from-zone trust to-zone untrust]
ibm@J58S-1# set policy in-out-1 match destination-address any

{primary:node0}[edit security policies from-zone trust to-zone untrust]
ibm@J58S-1# set policy in-out-1 match application any
```

```

{primary:node0}[edit security policies from-zone trust to-zone untrust]
ibm@J58S-1# set policy in-out-1 then permit

{primary:node0}[edit security policies from-zone trust to-zone untrust]
ibm@J58S-1# top

{primary:node0}[edit]
ibm@J4350-1# copy security policies from-zone trust to-zone untrust to from-zone untrust to-zone trust

{primary:node0}[edit]
ibm@J58S-1# edit security ike

{primary:node0}[edit security ike]
ibm@J58S-1# set proposal ike-prop1 authentication-method pre-shared-keys

{primary:node0}[edit security ike]
ibm@J58S-1# set proposal ike-prop1 dh-group group5

{primary:node0}[edit security ike]
ibm@J58S-1# set proposal ike-prop1 authentication-algorithm sha-256

{primary:node0}[edit security ike]
ibm@J58S-1# set proposal ike-prop1 encryption-algorithm aes-256-cbc

{primary:node0}[edit security ike]
ibm@J58S-1# set policy ike-pol1 mode main

{primary:node0}[edit security ike]
ibm@J58S-1# set policy ike-pol1 proposals ike-prop1

{primary:node0}[edit security ike]
ibm@J58S-1# set policy ike-pol1 pre-shared-key ascii-text vpn_key

{primary:node0}[edit security ike]
ibm@J58S-1# set gateway gw1 ike-policy ike-pol1

{primary:node0}[edit security ike]
ibm@J58S-1# set gateway gw1 address 10.10.20.20

{primary:node0}[edit security ike]
ibm@J58S-1# set gateway gw1 dead-peer-detection interval 10

{primary:node0}[edit security ike]
ibm@J58S-1# set gateway gw1 external-interface ge-1/2/0.0

{primary:node0}[edit security ike]
ibm@J58S-1# top edit security ipsec

{primary:node0}[edit security ipsec]
ibm@J58S-1# set proposal ipsec-prop1 protocol esp

{primary:node0}[edit security ipsec]
ibm@J58S-1# set proposal ipsec-prop1 authentication-algorithm hmac-sha1-96

```

```

{primary:node0}[edit security ipsec]
ibm@J58S-1# set proposal ipsec-prop1 encryption-algorithm aes-256-cbc

{primary:node0}[edit security ipsec]
ibm@J58S-1# set policy ipsec-pol1 proposals ipsec-prop1

{primary:node0}[edit security ipsec]
ibm@J58S-1# set vpn vpn1 bind-interface st0.0

{primary:node0}[edit security ipsec]
ibm@J58S-1# set vpn vpn1 ike gateway gw1

{primary:node0}[edit security ipsec]
ibm@J58S-1# set vpn vpn1 ike ipsec-policy ipsec-pol1

{primary:node0}[edit security ipsec]
ibm@J58S-1# set vpn vpn1 establish-tunnels immediately

{primary:node0}[edit security ipsec]
ibm@J58S-1# top

{primary:node0}[edit]
ibm@J58S-1# commit
node0:
commit complete

{primary:node0}[edit]
ibm@J58S-1# show security ike
proposal ike-prop1 {
    authentication-method pre-shared-keys;
    dh-group group5;
    authentication-algorithm sha-256;
    encryption-algorithm aes-256-cbc;
}
policy ike-pol1 {
    mode main;
    proposals ike-prop1;
    pre-shared-key ascii-text "$9$LKz7wg4aU.PQLx4ZUjPfy1K"; ## SECRET-DATA
}
gateway gw1 {
    ike-policy ike-pol1;
    address 10.10.20.20;
    dead-peer-detection interval 10;
    external-interface ge-1/2/0.0;
}

{primary:node0}[edit]
ibm@J58S-1# show security ipsec
proposal ipsec-prop1 {
    protocol esp;
    authentication-algorithm hmac-shal-96;
    encryption-algorithm aes-256-cbc;
}
policy ipsec-pol1 {
    proposals ipsec-prop1;

```



```

}
vpn vpn1 {
  bind-interface st0.0;
  ike {
    gateway gw1;
    ipsec-policy ipsec-pol1;
  }
  establish-tunnels immediately;
}

{primary:node0}[edit]
ibm@J58S-1# run show security ike security-associations
Index   Remote Address  State  Initiator cookie  Responder cookie  Mode
1       10.10.20.20     UP     11ab496a3e237c79  dc96be57b2758072  Main

{primary:node0}[edit]
ibm@J58S-1# run show security ipsec security-associations
Total active tunnels: 1
ID      Gateway      Port  Algorithm          SPI      Life:sec/kb  Mon vsys
<131073 10.10.20.20   500   ESP:aes-256/sha1   9f2b4205  28766/unlim  -   0
>131073 10.10.20.20   500   ESP:aes-256/sha1   22301610  28766/unlim  -   0

{primary:node0}[edit]
ibm@J58S-1# run show security ipsec statistics
ESP Statistics:
  Encrypted bytes:      11696
  Decrypted bytes:      7224
  Encrypted packets:    86
  Decrypted packets:    86
AH Statistics:
  Input bytes:          0
  Output bytes:         0
  Input packets:        0
  Output packets:       0
Errors:
  AH authentication failures: 0, Replay errors: 0
  ESP authentication failures: 0, ESP decryption failures: 0
  Bad headers: 0, Bad trailers: 0

```

8.6.4 IPsec dynamic VPN and PKI support

IBM j-type security appliances also supports Dynamic VPN, and PKI for VPN authentication.

Public key infrastructure refers to the hierarchical structure of trust required for the successful implementation of public key cryptography. To verify the trustworthiness of a certificate, you must be able to track a path of certified certificate authorities from the one issuing your local certificate to the root authority of a CA domain.

Dynamic VPN tunnels enable users to securely access assets such as email servers and application servers that reside behind a firewall. End-to-site VPN tunnels are particularly helpful to remote users such as telecommuters because a single tunnel enables access to all of the resources on a network—the users do not need to configure individual access settings to each application and server.

These options are not common for data center scenarios and are not covered in this book. For more information about these features for IBM j-type security appliances, refer to “*JUNOS Software Security Configuration Guide, Release 10.1*” book

8.7 Authentication options

This section presents an overview of authentication options and how to configure and monitor them on IBM j-type security appliances.

8.7.1 User authentication overview

A firewall user is a network user who must provide a username and password for authentication when initiating a connection across the firewall. Junos software enables administrators to restrict and permit firewall users to access protected resources (different zones) behind a firewall based on their source IP address and other credentials.

After you define firewall users, you can create a policy that requires the users to authenticate themselves through one of two authentication schemes:

- ▶ **Pass-through authentication:** A host or a user from one zone tries to access resources on another zone. You must use an FTP, a Telnet, or an HTTP client to access the IP address of the protected resource and to get authenticated by the firewall. The device uses FTP, Telnet, or HTTP to collect username and password information, and subsequent traffic from the user or host is allowed or denied based on the result of this authentication.
- ▶ **Web authentication:** Users try to connect, using HTTP, to an IP address on the device that is enabled for web authentication; in this scenario, you do not use HTTP to get to the IP address of the protected resource. You are prompted for the username and password that are verified by the device. Subsequent traffic from the user or host to the protected resource is allowed or denied based on the result of this authentication.

8.7.2 Pass-through authentication

With pass-through user authentication, when a user attempts to initiate an HTTP, an FTP, or a Telnet connection request that has a policy requiring authentication, the device intercepts the request and prompts the user to enter a username and password.

Before granting permission, the device validates the username and password by checking them against those stored in the local database or on an external authentication server.

After the device authenticates a user at a particular source IP address, it subsequently permits traffic—as specified in the policy requiring authentication through pass through—from any other user at that same address. This might be the case if the user originates traffic from behind a NAT device that changes all original source addresses to a single translated address.

To configure pass-through authentication:

1. Create IP addresses for the device interfaces.
2. Create an access profile for the user, and specify the password.
3. Add an authentication profile for pass-through firewall authentication.
4. Define a success banner for Telnet sessions.
5. Create security zones.
6. Assign the security policy to the zones.

To create an authentication profile for a client and assign a local password, use the command:

```
{primary:node0}[edit]
ibm@J58S-1# set access profile access_profile client client_name firewall-user
password client_password
```

In the command:

- ▶ *Access_profile* is the name of the access profile we are creating
- ▶ *Client_name* is the username of the client
- ▶ *Client_password* is the local password for this client

To add the authentication profile for pass-through firewall authentication, use the command:

```
{primary:node0}[edit]
ibm@J58S-1# set access firewall-authentication pass-through default-profile
access_profile
```

In the command, *access_profile* is the name of the access profile created before.

To configure a banner for successful authentication through telnet, use the commands:

```
{primary:node0}[edit]
ibm@J58S-1# set access firewall-authentication pass-through telnet banner success
"success banner"
```

To activate the pass-through firewall authentication policy for a particular security policy, use the command:

```
{primary:node0}[edit]
ibm@J58S-1# set security policies from-zone from-zone to-zone to-zone policy
policy_name then permit firewall-authentication pass-through client-match
access_profile
```

In the command:

- ▶ *From-zone* is the source zone
- ▶ *To-zone* is the destination zone
- ▶ *Policy_name* is the name of this policy
- ▶ *Access_profile* is the name of the access profile used for pass-through authentication

To verify firewall user authentication, use the command

```
{primary:node0}
ibm@J58S-1> show security firewall-authentication history
```

To monitor users and IP addresses in authentication table, use the command:

```
{primary:node0}
ibm@J58S-1> show security firewall-authentication history
```

8.7.3 Web authentication

Web authentication is an alternative to pass-through user authentication. Instead of pointing to the resource that you want to connect to from your client browser, you point the browser to an IP address on the device that is enabled for web authentication. This initiates an HTTP session to the IP address hosting the web authentication feature on the device. The device then prompts you for your username and password and caches the result in the device. Later, when traffic encounters a web authentication policy, you are allowed or denied access based on the prior web authentication results,

Follow these web authentication guidelines:

- ▶ You can leave the default web authentication server as the local database or you can choose an external authentication server for the role. The default web authentication profile determines if the user authenticates using the local database or the external authentication server. An access profile stores usernames and passwords of users or points to external authentication servers where such information is stored.
- ▶ The web authentication address must be in the same subnet as the interface that you want to use to host it.
- ▶ You can put a web authentication address in the same subnet as the IP address of any physical interface or virtual security interface (VSI).
- ▶ You can put web authentication addresses on multiple interfaces.
- ▶ After a device authenticates a user at a particular source IP address, it subsequently permits traffic—as specified in the policy requiring authentication through web authentication—from any other user at that same address. This might be the case if the user originates traffic from behind a NAT device that changes all original source addresses to a single translated address.
- ▶ With web authentication enabled, any HTTP traffic to the IP address will get the web authentication login page instead of the administrator login page. Disabling this option shows the administrator login page (assuming that [system services web-management HTTP] is enabled).
- ▶ We recommend that you have a separate primary or preferred IP address, if an address is used for web authentication.

To configure web authentication:

1. Create IP addresses for the device interfaces.
2. Create an access profile for the user and specify the password.
3. Add an authentication profile for firewall web authentication.
4. Create security zones.
5. Assign the security policy to the zones.
6. Activate the HTTP daemon on your device.

To enable an interface for web authentication, use the **web-authentication http** command in its logical unit configuration:

```
{primary:node0}[edit]
ibm@J58S-1# set interfaces interface family inet address IP_address/mask
web-authentication http
```

In the command:

- ▶ Interface is the name of the interface in type-fpc/pic/port.logical-unit-number notation
- ▶ IP_address/mask is the assigned IP address and mask of the interface

Activate http service on the appliance with the command:

```
{primary:node0}[edit]
ibm@J58S-1# set system services web-management http
```

Create an authentication profile for a client and assign a local password with the command:

```
{primary:node0}[edit]
ibm@J58S-1# set access profile access_profile client client_name firewall-user
password client_password
```

In the command:

- ▶ `Access_profile` is the name of the access profile we are creating
- ▶ `Client_name` is the username of the client
- ▶ `Client_password` is the local password for this client

To add the authentication profile for firewall web-authentication, use the command:

```
{primary:node0}[edit]
ibm@J58S-1# set access firewall-authentication web-authentication default-profile
access_profile
```

In the command, `access_profile` is the name of the access profile created before.

To activate the web-authentication firewall authentication policy for a particular security policy, use the command:

```
{primary:node0}[edit]
ibm@J58S-1# set security policies from-zone from-zone to-zone to-zone policy
policy_name then permit firewall-authentication web-authentication client-match
access_profile
```

In the command:

- ▶ `From-zone` is the source zone
- ▶ `To-zone` is the destination zone,
- ▶ `Policy_name` is the name of this policy
- ▶ `Access_profile` is the name of the access profile used for pass-through authentication

To verify firewall user authentication, use the command:

```
{primary:node0}
ibm@J58S-1> show security firewall-authentication history
```

To monitor users and IP addresses in authentication table, use the command:

```
{primary:node0}
ibm@J58S-1> show security firewall-authentication history
```

8.7.4 External authentication servers

AAA provides an extra level of protection and control for user access in the following ways:

- ▶ Authentication determines the firewall user.
- ▶ Authorization determines what the firewall user can do.
- ▶ Accounting determines what the firewall user did in the network.

You can use authentication alone or with authorization and accounting. Authorization always requires a user to be authenticated first. You can use accounting alone, or with authentication and authorization.

After the user's credentials are collected, they are processed using firewall user authentication, which supports the following types of servers:

- ▶ Local authentication and authorization
- ▶ RADIUS authentication and authorization (compatible with Funk RADIUS server)
- ▶ LDAP authentication only (supports LDAP version 3 and compatible with Windows AD)
- ▶ SecurID authentication only (using an RSA SecurID external authentication server)

Note: Local, RADIUS, LDAP, and RSA SecurID are only supported for firewall authentication for traffic of users. Administrators can use Local, RADIUS, and TACACS for device management.

Configure the appliance for external authentication with the command:

```
{primary:node0}[edit]
ibm@J58S-1# set access profile access_profile authentication-order options
```

In the command:

- ▶ Access_profile is the access profile for which we want external authentication
- ▶ Options can be a combination of one or multiple authentication methods, such as radius, ldap, securid, and password

Default behavior is to use local password database.

To restrict the firewall users to authenticate through the external server only, do not use the **password** option. If the external server authentication fails and the default password (local database) option is not specified, the firewall user is locked out.

To define a RADIUS server, use the command:

```
{primary:node0}[edit]
ibm@J58S-1# set access profile access_profile radius-server IP_address secret
radius_secret
```

In the command:

- ▶ Access_profile is the access profile for which we want external authentication
- ▶ IP_address is the IP address of the RADIUS server
- ▶ Radius_secret is the secret for the RADIUS server

To define a LDAP server, use the command:

```
{primary:node0}[edit]
ibm@J58S-1# set access profile access_profile ldap-server IP_address
```

In the command:

- ▶ Access_profile is the access profile for which we want external authentication
- ▶ IP_address is the IP address of the LDAP server

To define the LDAP server options, use the command:

```
{primary:node0}[edit]
ibm@J58S-1# set access profile access_profile ldap-options options
```

In the command:

- ▶ Access_profile is the access profile for which we want external authentication
- ▶ Options are the options of the LDAP server

For securid configuration, refer to the *JUNOS Software Security Configuration Guide, Release 10.1* at:

<http://www.juniper.net/techpubs/software/junos-security/junos-security10.1/junos-security-swconfig-security/junos-security-swconfig-security.pdf>

8.7.5 Client groups for firewall authentication

To manage a number of firewall users, you can create user or client groups and store the information either on the local IBM j-type security appliance or on an external RADIUS or LDAP server.

A client group is a list of groups to which the client belongs. As with client-idle timeout, a client group is used only if the external authentication server does not return a value in its response (for example, LDAP servers do not return such information).

The RADIUS server sends the client's group information to the IBM j-type security appliance using Juniper VSA (46). The client-match portion of the policy accepts a string that can either be the username or group name the client belongs to.

The reason to have a single database for different types of clients (except admins) is based on the assumption that a single client can be of multiple types. For example, a firewall user client can also be an L2TP client.

To configure a default client groups for all users from an access profile:

```
{primary:node0}[edit]
ibm@J58S-1# set access profile access_profile session-options client-group [
group1 groupN ]
```

In the command:

- ▶ Access_profile is the access profile for which we want client grouping
- ▶ Group1 to groupN are the groups assigned to this profile (can be one or more groups)

8.7.6 User authentication configuration example

In Example 8-8, we used the security zone definition from Example 8-2 on page 201 and security policy definition from Example 8-3 on page 212, and based on this preconfiguration we demonstrate additional authentication with the following characteristics:

1. Create an authentication profile with name ap1.
2. Add a client name user1 in this authentication profile.
3. Set the user1 password to user1pass.
4. Set the default pass-through profile to ap1.
5. Set the telnet login banner to Please enter your firewall username and password.
6. Set the telnet success banner to Authentication done.
7. Set the telnet fail banner to Authentication failed.
8. Modify policy in-out-2 from zone inside to zone outside that will permit all traffic if is pass-through authenticated with user1.

Example 8-8 Pass-through authentication configuration

```
{primary:node0}[edit]
ibm@J58S-1# set access profile ap1 client user1 firewall-user password user1pass

{primary:node0}[edit]
ibm@J58S-1# set access firewall-authentication pass-through default-profile ap1

{primary:node0}[edit]
```

```

ibm@J58S-1# set access firewall-authentication pass-through telnet banner login
"Please enter your firewall username and password"

{primary:node0}[edit]
ibm@J58S-1# set access firewall-authentication pass-through telnet banner success
"Authentication done"

{primary:node0}[edit]
ibm@J58S-1# set access firewall-authentication pass-through telnet banner fail
"Authentication failed"

{primary:node0}[edit]
ibm@J58S-1# set security policies from-zone inside to-zone outside policy in-out-2
match source-address any

{primary:node0}[edit]
ibm@J58S-1# set security policies from-zone inside to-zone outside policy in-out-2
match destination-address any

{primary:node0}[edit]
ibm@J58S-1# set security policies from-zone inside to-zone outside policy in-out-2
match application any

{primary:node0}[edit]
ibm@J58S-1# set security policies from-zone inside to-zone outside policy in-out-2
then permit firewall-authentication pass-through client-match user1

{primary:node0}[edit]
ibm@J58S-1# commit
node0:
commit complete

{primary:node0}[edit]
ibm@J58S-1# show access
profile ap1 {
  client user1 {
    firewall-user {
      password "$9$.PF/tu1SyK0BLNdV4oFn/tBE"; ## SECRET-DATA
    }
  }
}
firewall-authentication {
  pass-through {
    default-profile ap1;
    telnet {
      banner {
        login "Please enter your firewall username and password";
        success "Authentication done";
        fail "Authentication failed";
      }
    }
  }
}
}

{primary:node0}[edit]

```



```

ibm@J58S-1# show security policies from-zone inside to-zone outside
policy in-out-1 {
    match {
        source-address any;
        destination-address out-mail;
        application any;
    }
    then {
        permit;
        count;
    }
}
policy in-out-2 {
    match {
        source-address any;
        destination-address any;
        application any;
    }
    then {
        permit {
            firewall-authentication {
                pass-through {
                    client-match user1;
                }
            }
        }
    }
}

{primary:node0}[edit]
ibm@J58S-1#

```

In Example 8-9, we use the J08E switch, which is connected to the inside zone, to ping the outside test IP address of 10.10.10.103. It is not working because we are not yet authenticated. Then we telnet to this test IP address of 10.10.10.103, and see that the firewall intercepts the telnet session and initiates the pass-through authentication. First we use a dummy password to check the fail banner, and then we authenticate with the right password. In the end of the example we try again to ping the outside test IP address of 10.10.10.103 which now works.

Example 8-9 Testing the pass-through authentication

```

{master}
ibm@J08E-re0> ping 10.10.10.103
PING 10.10.10.103 (10.10.10.103): 56 data bytes
^C
--- 10.10.10.103 ping statistics ---
7 packets transmitted, 0 packets received, 100% packet loss

{master}
ibm@J08E-re0> telnet 10.10.10.103
Trying 10.10.10.103...
Connected to 10.10.10.103.
Escape character is '^]'.
Please enter your firewall username and password
Username: user1

```

```

Password:
      Authentication failed
Connection closed by foreign host.

{master}
ibm@J08E-re0> telnet 10.10.10.103
Trying 10.10.10.103...
Connected to 10.10.10.103.
Escape character is '^]'.
Please enter your firewall username and password
Username: user1
Password:
      Authentication done

```

J02M (ttyp0)

```

login: ^CClient aborted login
Connection closed by foreign host.

```

```

{master}
ibm@J08E-re0> ping 10.10.10.103
PING 10.10.10.103 (10.10.10.103): 56 data bytes
64 bytes from 10.10.10.103: icmp_seq=0 ttl=63 time=1.392 ms
64 bytes from 10.10.10.103: icmp_seq=1 ttl=63 time=1.220 ms
64 bytes from 10.10.10.103: icmp_seq=2 ttl=63 time=1.382 ms
64 bytes from 10.10.10.103: icmp_seq=3 ttl=63 time=1.128 ms
^C
--- 10.10.10.103 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.128/1.280/1.392/0.111 ms

{master}
ibm@J08E-re0>

```

In Example 8-10 we go back to the J58S security appliance to check for authenticated users:

Example 8-10 Check for authenticated users

```

{primary:node0}[edit]
ibm@J58S-1# run show security firewall-authentication users
node0:
-----
Firewall authentication data:
Total users in table: 1

```

	Id	Source Ip	Src zone	Dst zone	Profile	Age	Status	User
	7	192.168.1.10	inside	outside	apl	0	Success	user1

```

{primary:node0}[edit]
ibm@J58S-1# run show security firewall-authentication history
node0:
-----
History of firewall authentication data:
Authentications: 6

```

	Id	Source Ip	Date	Time	Duration	Status	User
	1	192.168.1.10	2010-05-05	12:14:26	0:10:15	Success	user1

8.8 Other security

This section presents a brief overview of other security options available on IBM j-type security appliances.

8.8.1 Attack detection and prevention

The IBM Intrusion Detection and Prevention (IDP) feature, also known as a stateful firewall, detects and prevents attacks in network traffic.

An exploit can be either an information-gathering probe or an attack to compromise, disable, or harm a network or network resource. In some cases, the distinction between the two objectives of an exploit can be unclear. For example, a barrage of TCP SYN segments might be an IP address sweep with the intent of triggering responses from active hosts, or it might be a SYN flood attack with the intent of overwhelming a network so that it can no longer function properly. Furthermore, because an attacker usually precedes an attack by performing reconnaissance on the target, we can consider information-gathering efforts as a precursor to an impending attack—that is, they constitute the first stage of an attack. Thus, the term exploit encompasses both reconnaissance and attack activities, and the distinction between the two is not always clear.

IBM provides various detection and defense mechanisms at the zone and policy levels to combat exploits at all stages of their execution:

- ▶ Screen options at the zone level.
- ▶ Firewall policies at the inter-, intra-, and super-zone policy levels (super-zone here means in global policies, where no security zones are referenced).

To secure all connection attempts, Junos software uses a dynamic packet-filtering method known as stateful inspection. Using this method, Junos software identifies various components in the IP packet and TCP segment headers—source and destination IP addresses, source and destination port numbers, and packet sequence numbers—and maintains the state of each TCP session and pseudo UDP session traversing the firewall. (Junos software also modifies session states based on changing elements such as dynamic port changes or session termination.) When a responding TCP packet arrives, Junos software compares the information reported in its header with the state of its associated session stored in the inspection table. If they match, the responding packet is allowed to pass the firewall. If the two do not match, the packet is dropped.

Junos software screen options secure a zone by inspecting, then allowing or denying, all connection attempts that require crossing an interface bound to that zone. Junos software then applies firewall policies, which can contain content filtering and IDP components, to the traffic that passes the screen filters.

Reconnaissance deterrence

Attackers can better plan their attack when they first know the layout of the targeted network (which IP addresses have active hosts), the possible entry points (which port numbers are active on the active hosts), and the constitution of their victims (which operating system the active hosts are running). To gain this information, attackers must perform reconnaissance.

IBM provides several screen options for deterring attackers' reconnaissance efforts and thereby hindering them from obtaining valuable information about the protected network and network resources.

Suspicious packet attributes

Attackers can craft packets to perform reconnaissance or launch denial-of-service attacks. Sometimes it is unclear what the intent of a crafted packet is, but the fact that it is crafted suggests that its being put to some kind of insidious use.

The following screen options can be enabled to block suspicious packets that might contain hidden threats:

- ▶ ICMP Fragment Protection
- ▶ Large ICMP Packet Protection
- ▶ Bad IP Option Protection
- ▶ Unknown Protocol Protection
- ▶ IP Packet Fragment Protection
- ▶ SYN Fragment Protection

DoS attack

The intent of a DoS attack is to overwhelm the targeted victim with a tremendous amount of bogus traffic so that the victim becomes so preoccupied processing the bogus traffic that legitimate traffic cannot be processed. The target can be the firewall, the network resources to which the firewall controls access, or the specific hardware platform or operating system of an individual host.

If a DoS attack originates from multiple source addresses, it is known as a distributed denial-of-service (DDoS) attack. Typically, the source address of a DoS attack is spoofed. The source addresses in a DDoS attack might be spoofed, or the actual addresses of compromised hosts might be used as “zombie agents” to launch the attack.

The device can defend itself and the resources it protects from DoS and DDoS attacks.

Basic screen configuration

To configure a screen, use the command:

```
{primary:node0}[edit]
ibm@J58S-1# set security screen ids-option screen_name options
```

In the command:

- ▶ *Screen_name* is the name of this screen policy
- ▶ *Options* are the security options we need to define for this screen

To bind a screen to a zone, use the command:

```
{primary:node0}[edit]
ibm@J58S-1# set security zones security-zone zone_name screen screen_name
```

In the command:

- ▶ *Zone_name* is the name of the zone on which we apply the screen
- ▶ *Screen_name* is the name of the screen policy

8.8.2 Transparent mode

Transparent mode affords the simplest way to add security to the network. In transparent mode, organizations can deploy a Juniper Networks firewall/VPN appliance without making any other changes to the network: firewall, VPN, IPS, and denial-of-service (DoS) mitigation functions work without an IP address, making the device “invisible” to the user.

On IBM j-type Ethernet Appliances, you can configure one or more bridge domains to perform Layer 2 bridging. A bridge domain is a set of logical interfaces that share the same flooding or broadcast characteristics. Like a virtual LAN (VLAN), a bridge domain spans one or more ports of multiple devices. Thus, the IBM j-type Ethernet Appliance can function as a Layer 2 switch with multiple bridge domains that participate in the same Layer 2 network.

In transparent mode, the IBM j-type Ethernet Appliance filters packets that traverse the device without modifying any of the source or destination information in the IP packet headers. Transparent mode is useful for protecting servers that mainly receive traffic from untrusted sources because there is no need to reconfigure the IP settings of routers or protected servers.

Transparent mode: Transparent mode is supported only for IPv4 traffic and the following security features are not supported in transparent mode:

- ▶ NAT is not supported.
- ▶ IPsec VPN is not supported.
- ▶ Application Layer Gateways (ALGs) and Intrusion Detection and Prevention (IDP) are not supported in this release.

In transparent mode, all physical ports on the device are assigned to Layer 2 interfaces. Do not route Layer 3 traffic through the device. Layer 2 zones can be configured to host Layer 2 interfaces, and security policies can be defined between Layer 2 zones. When packets travel between Layer 2 zones, security policies can be enforced on these packets.

A device operates in Layer 2 transparent mode when all physical interfaces on the device are configured as Layer 2 interfaces. A physical interface is a Layer 2 interface if its logical interface is configured with the bridge family.

There is no command to define or enable transparent mode on the device. The device operates in transparent mode when there are interfaces defined as Layer 2 interfaces. The device operates in route mode (the default mode) if there are no physical interfaces configured as Layer 2 interfaces.

Mode: At the time of writing, the IBM j-type Ethernet Appliance can operate at either route mode or transparent mode, but not both modes at the same time. Changing the mode requires a reboot of the device.

You can configure the fxp0 out-of-band management interface on the IBM j-type Ethernet Appliance as a Layer 3 interface, even if Layer 2 interfaces are defined on the device. With the exception of the fxp0 interface, you must not define Layer 2 and Layer 3 interfaces on the device’s network ports.



Advanced configuration

In this chapter, we discuss the advanced configuration of IBM j-type s-series Ethernet Appliance.

This chapter discusses the following topics:

- ▶ Chassis cluster
- ▶ Virtual Router

9.1 Chassis cluster

This section discusses the features of chassis cluster and followed by the techniques of building high availability of IBM j-type s-series Ethernet Appliance in different deployment scenarios.

9.1.1 Chassis cluster overview

Chassis clustering provides network node redundancy by grouping a pair of the same model of IBM Ethernet Appliance into a cluster. The devices must be running Junos Software. The chassis cluster consists of a primary node and a secondary standby node. These nodes back each other up in the event of software or hardware failures. Failover between nodes is stateful to ensure that established sessions are not dropped during the transition. The two nodes synchronize configuration, processes, and services across two Ethernet links—the control link forming the control plane and the fabric link forming the data plane.

Control plane

The control link connected between the two nodes form a control plane that synchronizes configuration and system processes to ensure high availability of the interfaces and services. The control plane software operates in active/backup mode. When configured as a chassis cluster, the two nodes back up each other, with one node acting as the primary device and the other as the secondary device, ensuring stateful failover of processes and services in the event of system or hardware failure. If the primary device fails, the secondary device takes over processing of traffic.

The control plane software:

- ▶ Runs the Routing Engine on the entire chassis cluster as well as the interfaces on both nodes.
- ▶ Manages system resources including the Packet Forwarding Engine (PFE) on either node.
- ▶ Synchronizes the configuration over the control link.
- ▶ Establishes and maintains sessions, including authentication, authorization, and accounting (AAA) functions.
- ▶ Manages application-specific signaling protocols.
- ▶ Establishes and maintains management sessions, such as Telnet connections.
- ▶ Handles asymmetric routing.
- ▶ Manages routing state, Address Resolution Protocol (ARP) processing, and Dynamic Host Configuration Protocol (DHCP) processing.

Control Links

The control link relies on a proprietary protocol to transmit session state, configuration, and liveness signals across the nodes.

On IBM J56S and IBM J58S devices, by default, all control ports are disabled. Each SPC in a device has two control ports, and each device can have multiple SPCs plugged into it. To set up the control link in a chassis cluster with IBM J56S or IBM J58S devices, you connect and configure the control ports that you will use on each device (fpcn and fpcn) and then initialize the device in cluster mode.

For IBM J34S and IBM J36S devices, there are dedicated chassis cluster (HA) control ports on the switch fabric board. No control link configuration is needed for IBM J34S and IB J36S devices.

Control link failure and recovery

The health of the control plane is monitored by heartbeats sent across the control link. These are sent at configured intervals and a failure is determined when lost heartbeats exceed a configured threshold value.

If the control link fails, the secondary standby node is disabled to prevent it from becoming active. By default, a disabled secondary device must be rebooted to recover the control link. Control link recovery can be configured to automatically reboot a disabled node to restore the control link.

Data plane

The connection over the fabric ports forms a data plane that manages traffic flow processing and session redundancy between the two nodes. Session or flow redundancy is synchronized between the nodes by special packets called runtime objects (RTOs) across the fabric data link. The data plane software operates in active/active mode. In a chassis cluster, session information is updated as traffic traverses either device and this information is transmitted between the nodes over the fabric link to guarantee that established sessions are not dropped when a failover occurs. In active/active mode, it is possible for traffic to ingress the cluster on one node and egress from the other node.

RTOs for synchronizing a session include:

- ▶ Session creation on RTOs for the first packet
- ▶ Session deletion and age out RTOs
- ▶ IPsec security associations (SAs)
- ▶ Change-related RTOs for timeout synchronization and TCP state changes

Data link failure and recovery

Because the data link maintains session synchronization, it is vital to the chassis cluster. Junos software sends fabric probes to determine data link failure events. On link failure, the secondary node is disabled and must be rebooted to recover. On completing reboot, the node synchronizes session states with RTOs from the primary node.

Redundancy groups

Redundancy groups are the concept in chassis clusters that defines which nodes are active or passive. A redundancy group is active on one node and backup on the other node. The redundancy group has a priority configured with the higher priority becoming active.

The following factors determine whether a redundancy group is primary:

- ▶ The priority configured for the node
- ▶ The node ID which is a factor in case the priority are tied
- ▶ The order in which the nodes come up
- ▶ Node preemption configuration

A lower priority node that boots up before a higher priority node claims primary node status. Node preemption can be configured to force one node's primacy after both nodes are booted up.

A chassis cluster can have multiple redundancy groups configured (from 1 to 128) in addition to the default redundancy group 0. Redundancy group 0 is always defined for the control

plane that controls the Routing Engine. For an active/ passive chassis cluster, there is a redundancy group 1 defined for the data plane.

Each redundancy group can failover independently and can be primary on only one node at a time. Each redundancy group (besides redundancy group 0) can be assigned one or more RETH interfaces whose state determines which node is primary for that redundancy group.

Redundant Ethernet interfaces

A redundant Ethernet interface is a pseudo interface that includes a physical interface from each node of the cluster. A redundant Ethernet interface can contain either a pair of Fast Ethernet interfaces or a pair of Gigabit Ethernet interfaces that are referred to as child interfaces of the redundant Ethernet interface (the redundant parent).

Each redundant Ethernet interface can contain only two interfaces because a cluster contains only two nodes. A redundant Ethernet interface's child interface is associated with the redundant Ethernet interface as part of the child interface configuration. The redundant Ethernet interface's child interface inherits most of its configuration from its parent.

A redundant Ethernet interface inherits its failover property from the redundancy group x to which it belongs. A redundant Ethernet interface remains active as long as its primary child interface is available/active.

9.1.2 Example: Active/passive chassis cluster deployment

In this section, you will configure an active/passive chassis cluster deployment. Active/passive chassis cluster is the most common type of chassis cluster deployment. It consists of two chassis members in a cluster. One of the chassis members actively provides routing, firewall, NAT, VPN and other security services, along with maintaining the control of chassis cluster. Another one of the chassis members passively maintains its state with active firewall for cluster failover capabilities when active chassis member becomes inactive.

Figure 9-1 on page 279 shows the physical topology of active/passive deployment

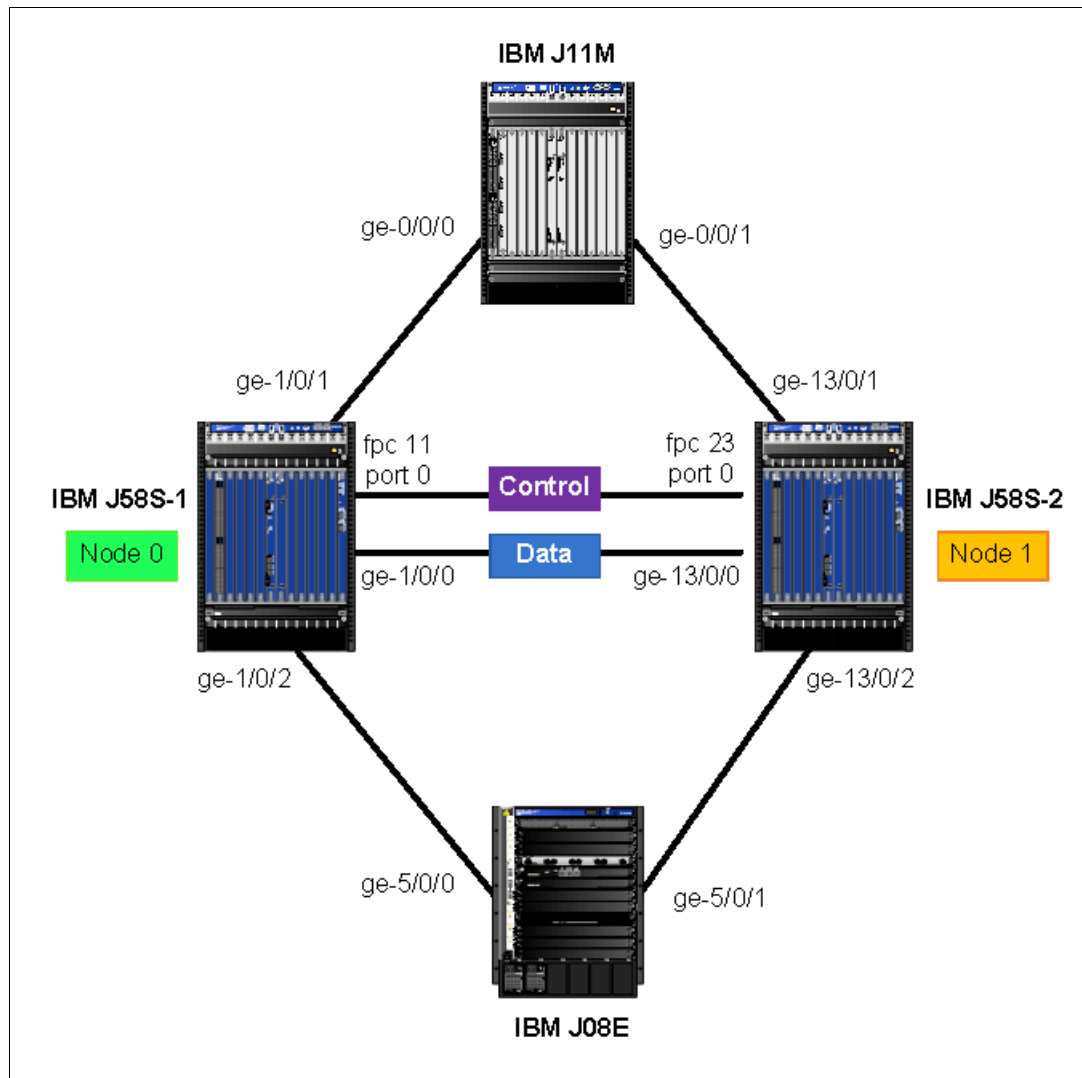


Figure 9-1 Physical topology of active/passive deployment

Figure 9-2 on page 280 shows the logical topology of active/passive deployment.

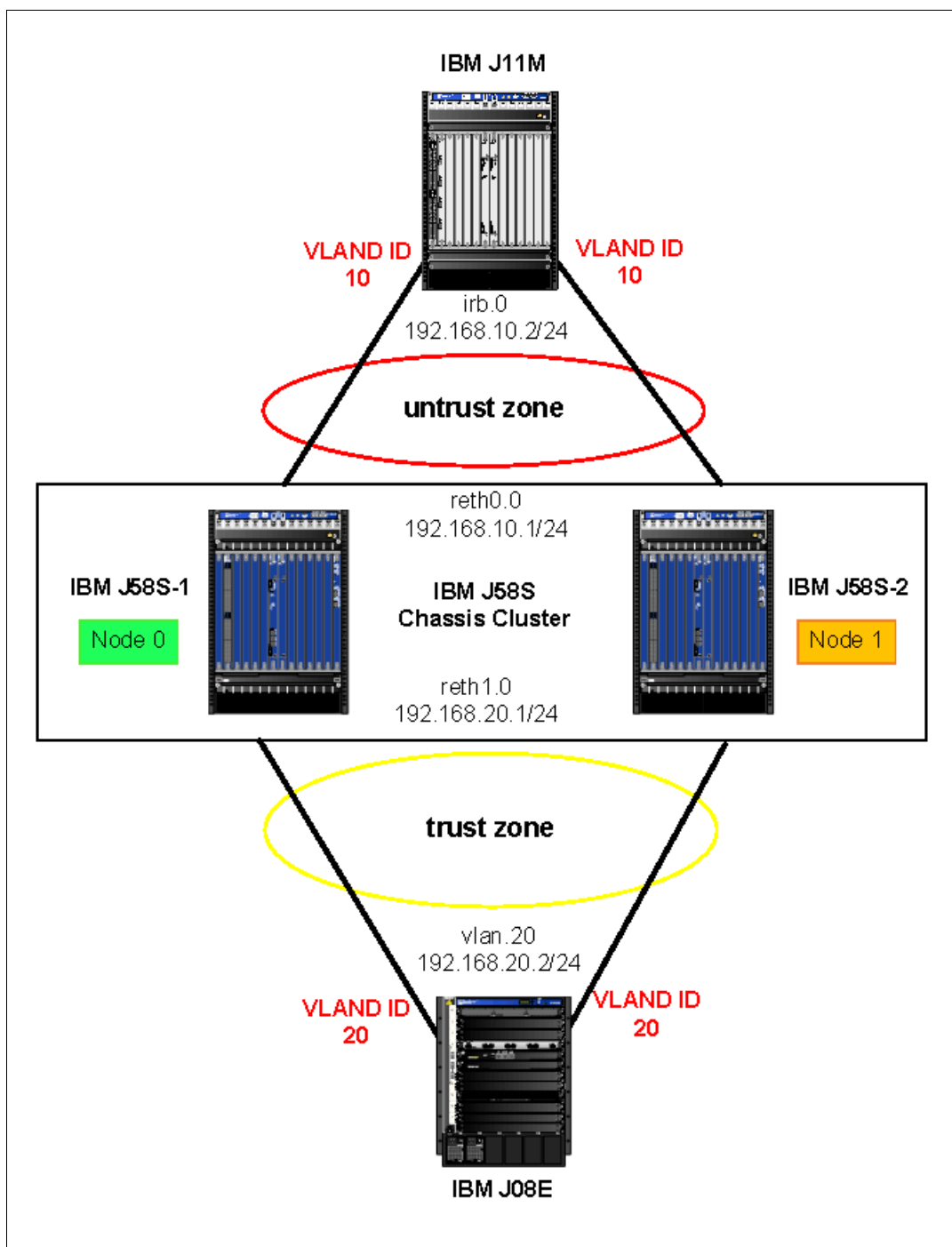


Figure 9-2 Logical topology of active/passive deployment

Before you start the configuration, you must ensure that both IBM Ethernet Appliance J58S have identical hardware chassis and are also running same version of Junos software.

Chassis cluster configuration

You begin the configuration by setting the cluster members to join the cluster. In this example, you only have a single cluster and set the cluster-id 1. IBM J58S-1 is assigned node 0 and IBM J58S-2 is node 1. This setting is configured on each of the chassis and it must be issued in operational mode not in configuration mode.

After keying this command, the cluster member must reboot, which is required at current Junos 10.0R1.8.

Example 9-1 shows the chassis cluster configuration applies to IBM J58S-1.

Example 9-1 Configure chassis cluster on IBM J58S-1

```
ibm@J58S-1> set chassis cluster cluster-id 1 node 0 reboot
Successfully enabled chassis cluster. Going to reboot now

ibm@J58S-1>
*** FINAL System shutdown message from root@J58S-1 ***
System going down IMMEDIATELY
```

Example 9-2 shows the chassis cluster configuration applies to IBM J58S-2.

Example 9-2 Configure chassis cluster on IBM J58S-2

```
ibm@J58S-2> set chassis cluster cluster-id 1 node 1 reboot
Successfully enabled chassis cluster. Going to reboot now

ibm@J58S-2>
*** FINAL System shutdown message from root@J58S-2 ***
System going down IMMEDIATELY
```

After each cluster member is rebooted, a node ID of each cluster member is displayed. To continue the configuration, you must issue the **configure shared** command to enter into configuration mode. Issuing the **configure** command is not allowed, and you are prompted with a warning. See Example 9-3.

Example 9-3 Enter configuration mode

```
{hold:node0}
ibm@J58S-1> configure
warning: Clustering enabled; using private edit
error: shared configuration database modified

Please temporarily use 'configure shared' to commit
outstanding changes in the shared database, exit,
and return to configuration mode using 'configure'

ibm@J58S-1> configure shared
Entering configuration mode
The configuration has been changed but not committed

{hold:node0}[edit]
```

Control port configuration

After the cluster members are rebooted, you move on to configure the control port of the clusters.

Before you begin to configure the control port, let us look at the slot numbering of chassis clustering. As a single system, the number slot of the chassis cluster is twice of a chassis member. In Figure 9-3, there are 24 slots where node 0 has the first twelve slot numbering and node 1 has the second twelve slot numbering.

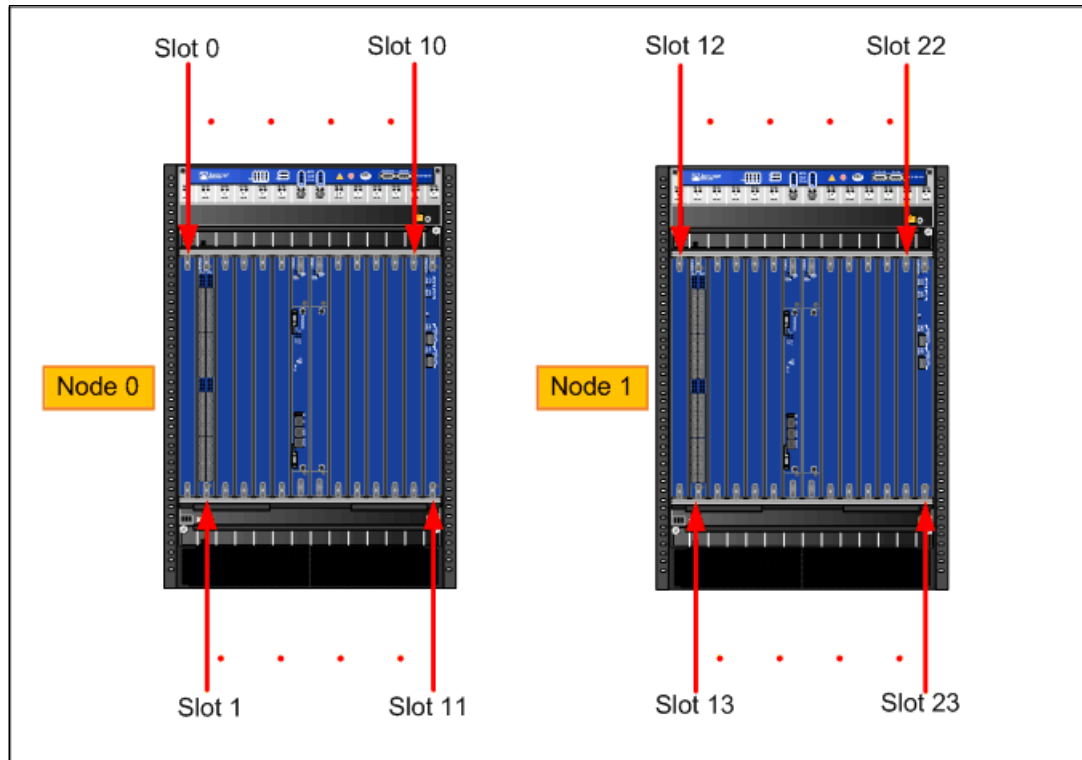


Figure 9-3 Slot numbering in a chassis clustering

With the Slot numbering information for both the nodes, you are ready to start the configuration of control port. Figure 9-4 on page 283 shows the slot and port number of Services Processing Card (SPC) that are used to interconnect both chassis members to establish a control link in this example.

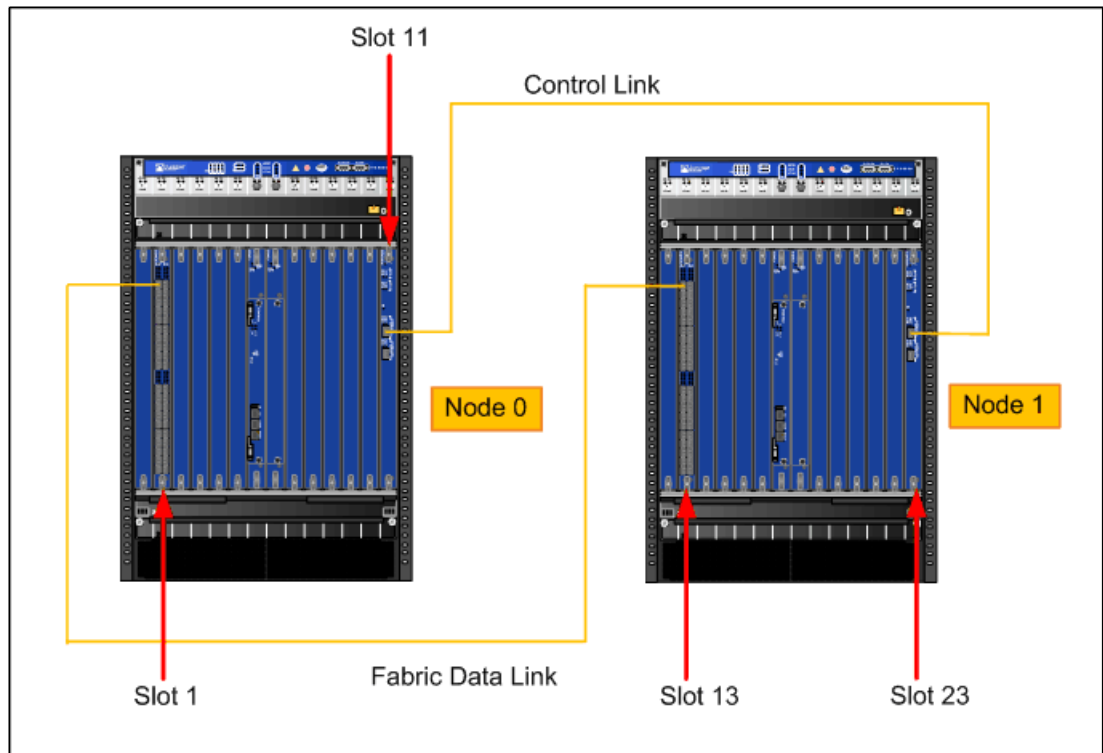


Figure 9-4 Slot and port number of Services Processing Card

With the slot and port number determined, you now configure the following commands to build a control link, as shown in Example 9-4.

Example 9-4 Configure control link

```
{hold:node0}[edit]
ibm@J58S-1#
set chassis cluster control-ports fpc 11 port 0
set chassis cluster control-ports fpc 23 port 0
```

Data fabric configuration

Now that the control ports are configured, you can now configure the data fabric ports of the cluster. These ports are used to pass runtime objects (RTOs) between cluster members. In this active/passive example, two 1Gb Ethernet ports are used in each chassis as it will not be using much bandwidth. However, 10Gb Ethernet ports can be used in active/active mode when high bandwidth is required. The physical port is shown in Figure 9-4, and the following configuration creates two data fabric ports, each one at a node.

Example 9-5 illustrates the configuration of data fabric ports.

Example 9-5 Configure data fabric ports

```
{hold:node0}[edit]
ibm@J58S-1#
set interfaces fab0 fabric-options member-interfaces ge-1/0/0
set interfaces fab1 fabric-options member-interfaces ge-13/0/0

ibm@J58S-1# show interfaces fab0
```

```

fabric-options {
    member-interfaces {
        ge-1/0/0;
    }
}

{primary:node0}[edit]
ibm@J58S-1# show interfaces fab1
fabric-options {
    member-interfaces {
        ge-13/0/0;
    }
}

```

Node-specific configuration

When both cluster members join together in a cluster, there is only a common configuration where you will only have access to the primary node in the cluster. To configure some specific setting to a member only, you can create a group to assign node specific configuration. In this example, you will create two separate IP address on fxp0 interface (out of band management) of each cluster member, as shown in Example 9-6.

Example 9-6 Configure node specific setting

```

{primary:node0}[edit]
ibm@J58S-1#
set groups node0 system host-name J58S-1
set groups node0 system backup-router 10.1.1.254
set groups node0 system backup-router destination 0.0.0.0/0
set groups node0 interfaces fxp0 unit 0 family inet address 10.1.1.12/24
set groups node1 system host-name J58S-2
set groups node1 system backup-router 10.1.1.254
set groups node1 system backup-router destination 0.0.0.0/0
set groups node1 interfaces fxp0 unit 0 family inet address 10.1.1.13/24

{primary:node0}[edit]
ibm@J58S-1# show groups
node0 {
    system {
        host-name J58S-1;
        backup-router 10.1.1.254 destination 0.0.0.0/0;
    }
    interfaces {
        fxp0 {
            unit 0 {
                family inet {
                    address 10.1.1.12/24;
                }
            }
        }
    }
}
node1 {
    system {
        host-name J58S-2;
    }
}

```



```

        backup-router 10.1.1.254 destination 0.0.0.0/0;
    }
    interfaces {
        fxp0 {
            unit 0 {
                family inet {
                    address 10.1.1.13/24;
                }
            }
        }
    }
}

```

Redundancy group configuration

Redundancy group 0 is always used for the control plane and redundancy group 1+n is normally used for the data plane ports.

Because this example is running in active/passive mode, only one chassis member is active at a time. Thus, you only define two Redundancy groups that are 0 and 1.

The priority of each node on a redundancy group also needs to be set as to determine which node is active for a redundancy group. You define node 0 has higher priority than node 1 for redundancy group 0 and 1 to make node 0 an active chassis for both control plane and data plane. See Example 9-7.

Example 9-7 Configuring the redundancy group

```

{primary:node0}[edit]
ibm@J58S-1#
set chassis cluster reth-count 2
set chassis cluster redundancy-group 0 node 0 priority 129
set chassis cluster redundancy-group 0 node 1 priority 128
set chassis cluster redundancy-group 1 node 0 priority 129
set chassis cluster redundancy-group 1 node 1 priority 128

{primary:node0}[edit]
ibm@J58S-1# show chassis
cluster {
    reth-count 2;
    redundancy-group 0 {
        node 0 priority 129;
        node 1 priority 128;
    }
    redundancy-group 1 {
        node 0 priority 129;
        node 1 priority 128;
    }
}

```

Figure 9-5 on page 286 shows the redundancy group and redundant Ethernet diagram.

Redundancy Group 1 (RG1) with Redundant Ethernet 0 (RETH0) and 1 (RETH1)

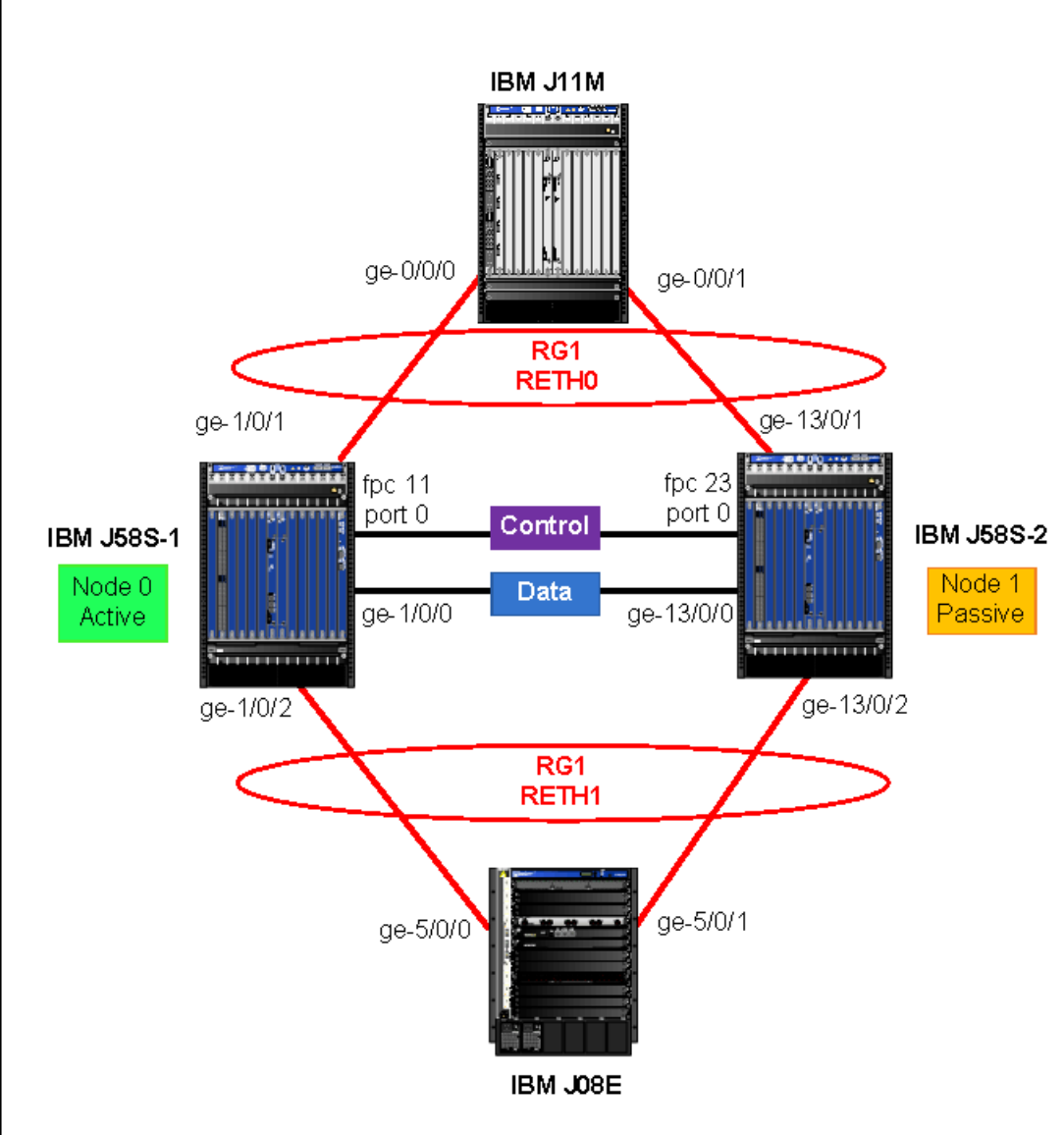


Figure 9-5 Redundancy group and redundant Ethernet diagram

Redundant Ethernet configuration

You now configure the actual interfaces on each chassis to bind to a Redundant Ethernet (RETH) interface. The RETH interface usually contains at least one interface from a chassis so as enabling the failover between chassis. This is similar to the aggregated Ethernet features.

Besides that, you need to assign the RETH interface to a Redundancy group. The RETH interface is also configured an IP address, as shown in Example 9-8.

Example 9-8 Configure redundant Ethernet

```
{primary:node0}[edit]
ibm@J58S-1#
set interfaces ge-1/0/1 gigether-options redundant-parent reth0
set interfaces ge-1/0/2 gigether-options redundant-parent reth1
```

```
set interfaces ge-13/0/1 gigether-options redundant-parent reth0  
set interfaces ge-13/0/2 gigether-options redundant-parent reth1
```

```
set interfaces reth0 redundant-ether-options redundancy-group 1  
set interfaces reth0 unit 0 family inet address 192.168.10.1/24  
set interfaces reth1 redundant-ether-options redundancy-group 1  
set interfaces reth1 unit 0 family inet address 192.168.20.1/24
```

```
{primary:node0}[edit]  
ibm@J58S-1# show interfaces ge-1/0/1  
gigether-options {  
    redundant-parent reth0;  
}
```

```
{primary:node0}[edit]  
ibm@J58S-1# show interfaces ge-1/0/2  
gigether-options {  
    redundant-parent reth1;  
}
```

```
{primary:node0}[edit]  
ibm@J58S-1# show interfaces ge-13/0/1  
gigether-options {  
    redundant-parent reth0;  
}
```

```
{primary:node0}[edit]  
ibm@J58S-1# show interfaces ge-13/0/2  
gigether-options {  
    redundant-parent reth1;  
}
```

```
{primary:node0}[edit]  
ibm@J58S-1# show interfaces reth0  
redundant-ether-options {  
    redundancy-group 1;  
}  
unit 0 {  
    family inet {  
        address 192.168.10.1/24;  
    }  
}
```

```
{primary:node0}[edit]  
ibm@J58S-1# show interfaces reth1  
redundant-ether-options {  
    redundancy-group 1;  
}  
unit 0 {  
    family inet {  
        address 192.168.20.1/24;  
    }  
}
```

Chassis interface monitoring

At this stage, you defined the cluster setting. Next, you configure how the cluster must behave in failures. All four Ethernet interfaces in this example are set to a failover threshold of 255 (weight). When the threshold reaches 0, it fails over to the other node.

You also configure the control link recovery which automatically causes the secondary node to reboot if the control link fails and then comes back up.

This is the final step for the chassis cluster configuration, as shown in Example 9-9.

Example 9-9 Configure chassis interface monitoring

```
{primary:node0}[edit]
ibm@J58S-1#
set chassis cluster control-link-recovery
set chassis cluster redundancy-group 1 interface-monitor ge-1/0/1 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-1/0/2 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-13/0/2 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-13/0/1 weight 255

{primary:node0}[edit]
ibm@J58S-1# show chassis cluster
control-link-recovery;
reth-count 2;
redundancy-group 1 {
    interface-monitor {
        ge-1/0/1 weight 255;
        ge-1/0/2 weight 255;
        ge-13/0/2 weight 255;
        ge-13/0/1 weight 255;
    }
}
```

Zone configuration

With the chassis cluster configuration complete, the rest of the configuration is exactly the same configuration as a stand-alone IBM Ethernet Appliance deployment. Each RETH interface must be assigned to the appropriate zone and virtual router and inbound traffic to the host. In this case, all system services and protocol are allowed for testing. See Example 9-10.

Example 9-10 Configure zone and inbound traffic

```
{primary:node0}[edit]
ibm@J58S-1#
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces reth0.0
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces reth1.0

{primary:node0}[edit]
ibm@J58S-1# show security zones security-zone untrust
host-inbound-traffic {
    system-services {
```

```

        all;
    }
    protocols {
        all;
    }
}
interfaces {
    reth0.0;
}

{primary:node0}[edit]
ibm@J58S-1# show security zones security-zone trust
host-inbound-traffic {
    system-services {
        all;
    }
    protocols {
        all;
    }
}
interfaces {
    reth1.0;
}

```

IBM J11M configuration

There are two ports on IBM J11M that are connected from IBM-J58S. These ports are configured as Ethernet bridging ports and set to a bridge domain with a VLAN ID 10. An Integrated Routing and Bridging (IRB) interface is created with an IP address, as shown in Example 9-11.

Example 9-11 Configure IBM J11M

```

{master}[edit]
ibm@J11M-re0#
set interfaces ge-0/0/0 encapsulation ethernet-bridge
set interfaces ge-0/0/0 unit 0 family bridge
set interfaces ge-0/0/1 encapsulation ethernet-bridge
set interfaces ge-0/0/1 unit 0 family bridge
set interfaces irb unit 0 family inet address 192.168.10.2/24

set bridge-domains To-J58S domain-type bridge
set bridge-domains To-J58S vlan-id 10
set bridge-domains To-J58S interface ge-0/0/0.0
set bridge-domains To-J58S interface ge-0/0/1.0
set bridge-domains To-J58S routing-interface irb.0

{master}[edit]
ibm@J11M-re0# show interfaces ge-0/0/0
encapsulation ethernet-bridge;
unit 0 {
    family bridge;
}

{master}[edit]

```

```

ibm@J11M-re0# show interfaces ge-0/0/1
encapsulation ethernet-bridge;
unit 0 {
    family bridge;
}

{master}[edit]
ibm@J11M-re0# show interfaces irb
unit 0 {
    family inet {
        address 192.168.10.2/24;
    }
}

{master}[edit]
ibm@J11M-re0# show bridge-domains
To-J58S {
    domain-type bridge;
    vlan-id 10;
    interface ge-0/0/0.0;
    interface ge-0/0/1.0;
    routing-interface irb.0;
}

```

IBM J08E configuration

Similar to J11M, two ports of IBM J58s are connected by IBM J08E. These ports are configured with a VLAN ID 20 and as access ports in layer 2. Then, VLAN ID 20 is bound with a layer 3 interface and set an IP address, as shown in Example 9-12.

Example 9-12 Configure IBM J08E

```

{master}[edit]
ibm@J08E-re0#
set vlans To-J58S vlan-id 20
set vlans To-J58S l3-interface vlan.20

set interfaces ge-5/0/0 unit 0 family ethernet-switching port-mode access
set interfaces ge-5/0/0 unit 0 family ethernet-switching vlan members To-J58S
set interfaces ge-5/0/1 unit 0 family ethernet-switching port-mode access
set interfaces ge-5/0/1 unit 0 family ethernet-switching vlan members To-J58S
set interfaces vlan unit 20 family inet address 192.168.20.2/24

{master}[edit]
ibm@J08E-re0# show vlans To-J58S
vlan-id 20;
l3-interface vlan.20;

{master}[edit]
ibm@J08E-re0# show interfaces ge-5/0/0
unit 0 {
    family ethernet-switching {
        port-mode access;
        vlan {
            members To-J58S;
        }
    }
}

```

```

    }
}

{master}[edit]
ibm@J08E-re0# show interfaces ge-5/0/1
unit 0 {
    family ethernet-switching {
        port-mode access;
        vlan {
            members To-J58S;
        }
    }
}

{master}[edit]
ibm@J08E-re0# show interfaces vlan.20
family inet {
    address 192.168.20.2/24;
}

```

Configuration verification

Junos software provides few commands to verify the working condition of chassis cluster.

Verifying the chassis cluster status

To display the failover status of a chassis cluster, use the **show chassis cluster status** command, as shown in Example 9-13.

Example 9-13 Verify chassis cluster status

```

{primary:node0}
ibm@J58S-1> show chassis cluster status
Cluster ID: 1
Node                Priority      Status    Preempt  Manual failover

Redundancy group: 0 , Failover count: 1
node0                129        primary   no       no
node1                128        secondary no       no

Redundancy group: 1 , Failover count: 1
node0                129        primary   no       no
node1                128        secondary no       no

```

The output shows failover status of the chassis cluster in addition to information about the chassis cluster redundancy groups.

Verifying chassis cluster statistics

To display information about the chassis cluster services and interfaces, use the **show chassis cluster statistics** command, as shown in Example 9-14.

Example 9-14 Verify chassis cluster statistics

```

ibm@J58S-1> show chassis cluster statistics
Control link statistics:

```

```

Control link 0:
  Heartbeat packets sent: 180681
  Heartbeat packets received: 180439
  Heartbeat packet errors: 0
Control link 1:
  Heartbeat packets sent: 0
  Heartbeat packets received: 0
  Heartbeat packet errors: 0
Fabric link statistics:
  Probes sent: 180673
  Probes received: 178156
  Probe errors: 0
Services Synchronized:
  Service name                RTOs sent    RTOs received
  Translation context          0             0
  Incoming NAT                 0             0
  Resource manager             0             0
  Session create               0             10
  Session close                0             10
  Session change               0              4
  Gate create                  0             0
  Session ageout refresh requests 0             0
  Session ageout refresh replies 0             0
  IPSec VPN                   0             0
  Firewall user authentication 0             0
  MGCP ALG                    0             0
  H323 ALG                    0             0
  SIP ALG                     0             0
  SCCP ALG                    0             0
  PPTP ALG                    0             0
  RPC ALG                     0             0
  RTSP ALG                    0             0
  RAS ALG                     0             0
  MAC address learning         0             0
  GPRS GTP                     0             0

{primary:node0}

```

The output shows the control link statistics (heartbeats sent and received), the fabric link statistics (probes sent and received) and the number of RTOs sent and received for services.

Verify chassis cluster interfaces

To display information about chassis cluster interfaces, use the **show chassis cluster interfaces** command, as shown in Example 9-15.

Example 9-15 Verify chassis cluster interfaces

```

{primary:node0}
ibm@J58S-1> show chassis cluster interfaces
Control link 0 name: em0
Control link 1 name: em1

Redundant-ethernet Information:
  Name           Status           Redundancy-group

```


reth0	Up	1
reth1	Up	1

Interface Monitoring:

Interface	Weight	Status	Redundancy-group
ge-13/0/1	255	Up	1
ge-13/0/2	255	Up	1
ge-1/0/2	255	Up	1
ge-1/0/1	255	Up	1

The output shows the state of the redundant Ethernet interfaces and the status of interfaces the are being monitored.

9.1.3 Example: Active/active chassis cluster deployment

Active/active chassis cluster deployment on IBM Ethernet Appliance J58S allow traffic to enter and exit to/from any direction on the cluster members. This deployment maintains traffic on both cluster members whenever possible.

In active/active deployment, only the data plane is functioning in active/active mode while the control plane is running in active/passive mode. The design allows one control plane to control both chassis member as a single logical device. In case of control plane failure, the control plane fails over to the secondary unit. This allows the data plane to fail over independently of the control plane.

When the traffic enters one cluster member and then exits a second cluster member when the best egress path for the traffic is out of the second cluster member, it is called Z-mode deployment which is typically characterized as an active/active deployment. This is an excellent design for extra throughput and redundancy. It is often used in conjunction with dynamic routing protocols which trigger failover to other cluster member when needed.

Figure 9-6 on page 294 shows the physical topology of active/active deployment.

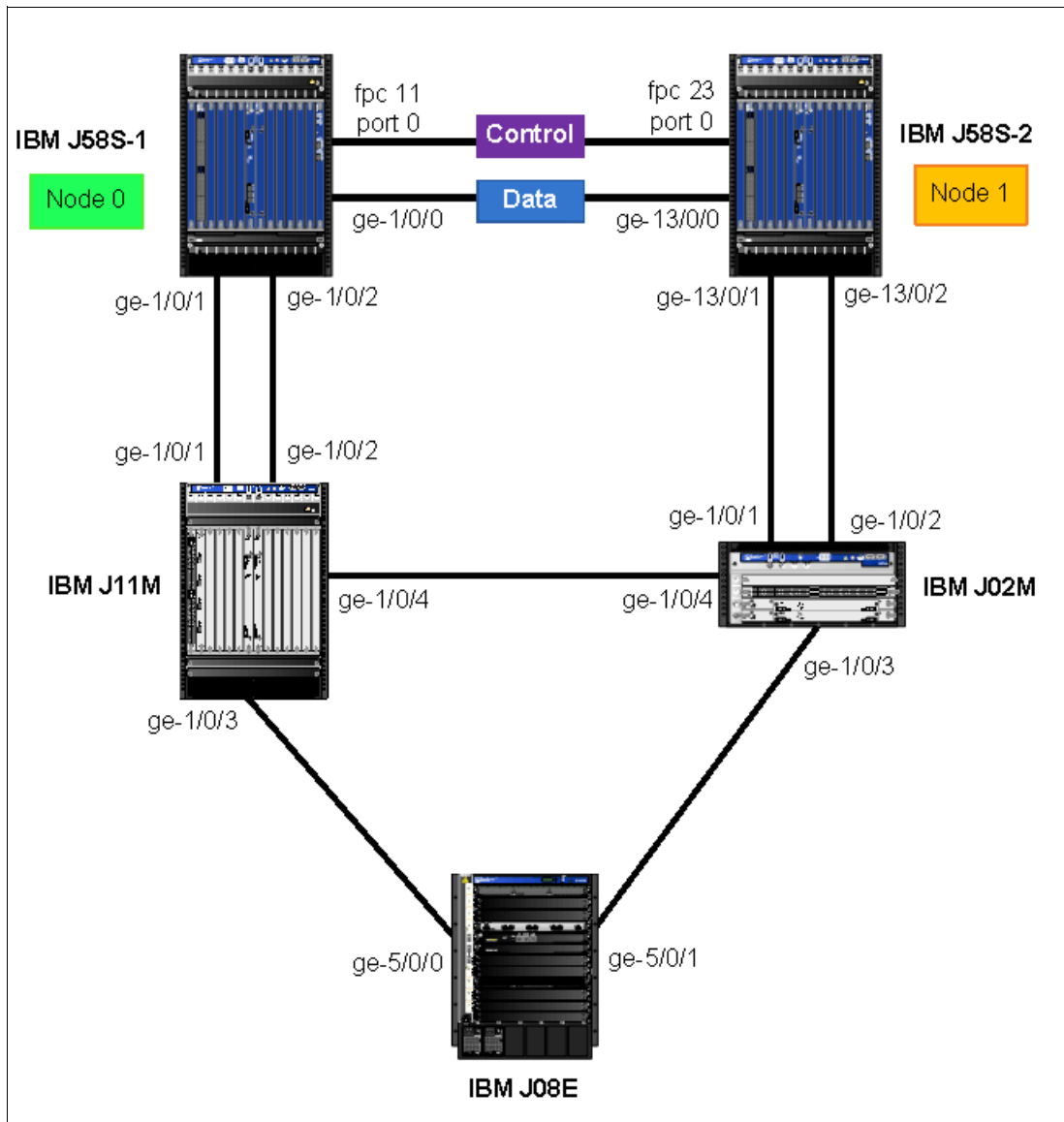


Figure 9-6 Physical topology of active/active deployment

Figure 9-7 on page 295 shows the logical topology of active/active deployment.

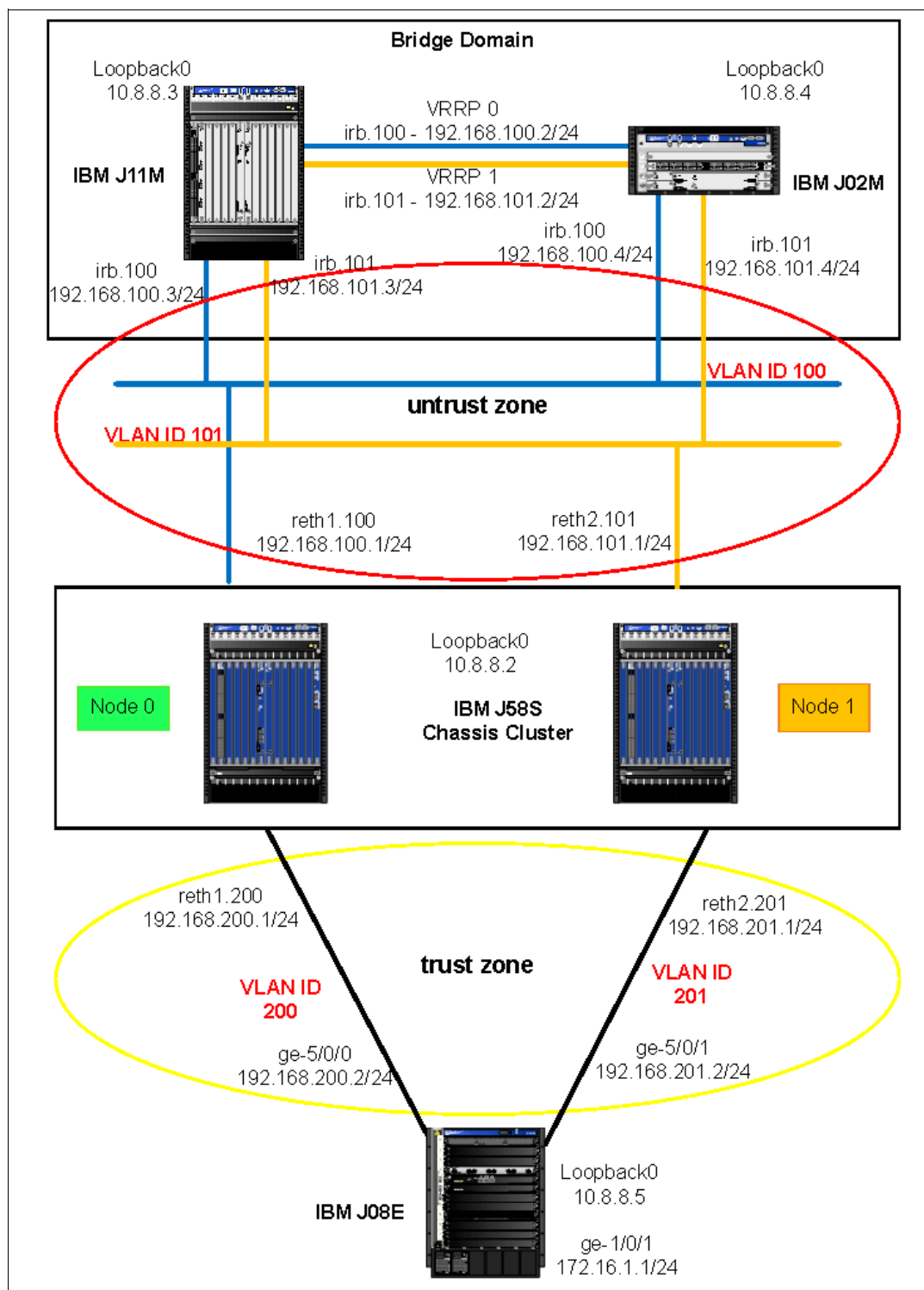


Figure 9-7 Logical topology of active/active deployment

Chassis cluster configuration

The configuration of this section is the same as in the active/passive chassis cluster deployment example that we discussed previously. You can refer to “Chassis cluster configuration” on page 280 for the configuration.

Control port configuration

The configuration of this section is the same as in the active/passive chassis cluster deployment example that we previously discussed. You can refer to “Control port configuration” on page 281 for the configuration.

Data fabric configuration

The configuration of this section is the same as in the active/passive chassis cluster deployment example that we previously discussed. You can refer to “Data fabric configuration” on page 283 for the configuration.

Node specific configuration

The configuration of this section is the same as in the active/passive chassis cluster deployment example that we previously discussed. You can refer to “Node-specific configuration” on page 284 for the configuration.

Redundancy group configuration

Redundancy group 0 is always used for the control plane and redundancy group 1+n is normally used for the data plane ports.

Because this example is running in active/active mode, you will create three redundancy groups. Redundancy group 0 is for the control plane. Redundancy group 1 and 2 are for the data plane.

The priority of each node on a redundancy group also needs to be set as to determine which node is active for a redundancy group. In this case, node 0 will have priority 254 and be active for redundancy group 0. Redundancy group 1 will be active on node 0 with priority 254. Conversely, redundancy group 2 will be active node 1, as shown in Example 9-16.

Example 9-16 Configure redundancy group

```
{primary:node0}[edit]
ibm@J58S-1#
set chassis cluster reth-count 3
set chassis cluster redundancy-group 0 node 0 priority 254
set chassis cluster redundancy-group 0 node 1 priority 128
set chassis cluster redundancy-group 1 node 0 priority 254
set chassis cluster redundancy-group 1 node 1 priority 128
set chassis cluster redundancy-group 1 preempt
set chassis cluster redundancy-group 2 node 0 priority 128
set chassis cluster redundancy-group 2 node 1 priority 254
set chassis cluster redundancy-group 2 preempt

{primary:node0}[edit]
ibm@J58S-1# show chassis cluster
reth-count 3;
redundancy-group 0 {
    node 0 priority 254;
    node 1 priority 128;
}
redundancy-group 1 {
    node 0 priority 254;
    node 1 priority 128;
    preempt;
}
redundancy-group 2 {
```

```

node 0 priority 128;
node 1 priority 254;
preempt;
}

```

Figure 9-8 shows the redundancy group and redundant Ethernet diagram.

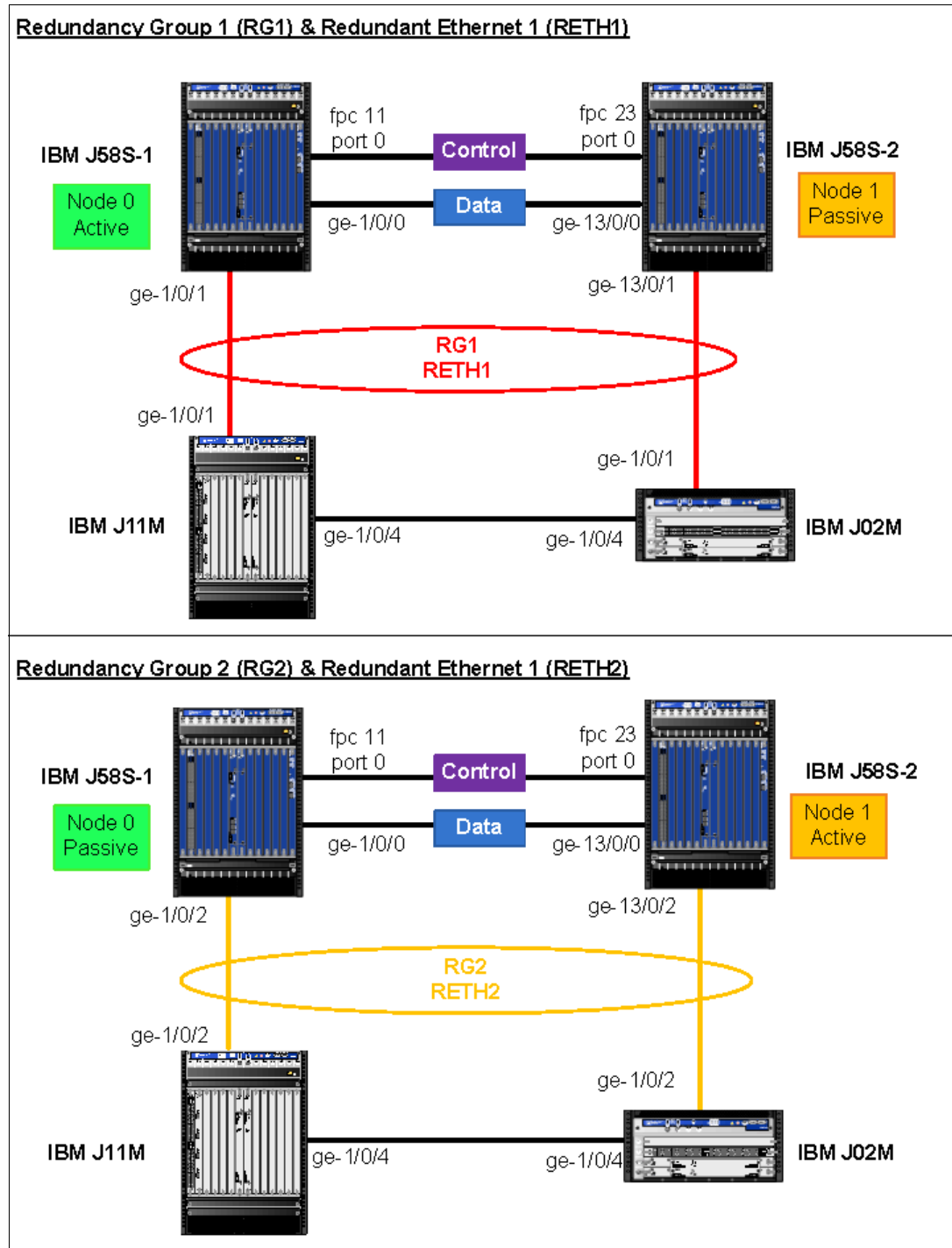


Figure 9-8 Redundancy group and redundant Ethernet diagram

Redundant Ethernet configuration

You now configure the actual interfaces on each chassis to bind to a Redundant Ethernet (RETH) interface. The RETH interface usually contains at least one interface from a chassis so as enabling the failover between chassis. This is similar to the aggregated Ethernet features

Besides that, you must assign the RETH interface to a redundancy group. The RETH interface is also configured an IP address. In this example, a physical interface carries multiple networks. Therefore, the interface is tagged with a VLAN to differentiate each network, as shown in Example 9-17.

Example 9-17 Configure redundant Ethernet

```
{primary:node0}[edit]
ibm@J58S-1#
set interfaces ge-1/0/1 gigether-options redundant-parent reth1
set interfaces ge-1/0/2 gigether-options redundant-parent reth2
set interfaces ge-13/0/1 gigether-options redundant-parent reth1
set interfaces ge-13/0/2 gigether-options redundant-parent reth2

set interfaces reth1 vlan-tagging
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 unit 100 vlan-id 100
set interfaces reth1 unit 100 family inet address 192.168.100.1/24
set interfaces reth1 unit 200 vlan-id 200
set interfaces reth1 unit 200 family inet address 192.168.200.1/24

set interfaces reth2 vlan-tagging
set interfaces reth2 redundant-ether-options redundancy-group 2
set interfaces reth2 unit 101 vlan-id 101
set interfaces reth2 unit 101 family inet address 192.168.101.1/24
set interfaces reth2 unit 201 vlan-id 201
set interfaces reth2 unit 201 family inet address 192.168.201.1/24

{primary:node0}[edit]
ibm@J58S-1# show interfaces ge-1/0/1
gigether-options {
    redundant-parent reth1;
}

{primary:node0}[edit]
ibm@J58S-1# show interfaces ge-1/0/2
gigether-options {
    redundant-parent reth2;
}

{primary:node0}[edit]
ibm@J58S-1# show interfaces ge-13/0/1
gigether-options {
    redundant-parent reth1;
}

{primary:node0}[edit]
ibm@J58S-1# show interfaces ge-13/0/2
gigether-options {
    redundant-parent reth2;
```

```

}

{primary:node0}[edit]
ibm@J58S-1# show interfaces reth1
vlan-tagging;
redundant-ether-options {
    redundancy-group 1;
}
unit 100 {
    vlan-id 100;
    family inet {
        address 192.168.100.1/24;
    }
}
unit 200 {
    vlan-id 200;
    family inet {
        address 192.168.200.1/24;
    }
}

{primary:node0}[edit]
ibm@J58S-1# show interfaces reth2
vlan-tagging;
redundant-ether-options {
    redundancy-group 2;
}
unit 101 {
    vlan-id 101;
    family inet {
        address 192.168.101.1/24;
    }
}
unit 201 {
    vlan-id 201;
    family inet {
        address 192.168.201.1/24;
    }
}

```

Chassis interface monitoring

At this stage, you defined the cluster setting. Next, configure how the cluster must behave in failures. All four Ethernet interfaces in this example are set to a failover threshold of 255 (weight). When the threshold reaches 0, it fails over to the other node.

You also configure the control link recovery that automatically causes the secondary node to reboot if the control link fails and then comes back up.

This is the final step in the chassis cluster configuration, as shown Example 9-18.

Example 9-18 Configure chassis interface monitoring

```

{primary:node0}[edit]
ibm@J58S-1#

```

```

set chassis cluster control-link-recovery
set chassis cluster redundancy-group 1 interface-monitor ge-1/0/1 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-13/0/1 weight 255
set chassis cluster redundancy-group 2 interface-monitor ge-1/0/2 weight 255
set chassis cluster redundancy-group 2 interface-monitor ge-13/0/2 weight 255

```

```

{primary:node0}[edit]
ibm@J58S-1# show chassis cluster
control-link-recovery;
redundancy-group 1 {
    node 0 priority 254;
    node 1 priority 128;
    preempt;
    interface-monitor {
        ge-1/0/1 weight 255;
        ge-13/0/1 weight 255;
    }
}
redundancy-group 2 {
    node 0 priority 128;
    node 1 priority 254;
    preempt;
    interface-monitor {
        ge-1/0/2 weight 255;
        ge-13/0/2 weight 255;
    }
}

```

Zone configuration

With chassis cluster configurations complete, the rest of the configuration is exactly the same configuration as a stand-alone IBM Ethernet Appliance deployment. Each RETH interface must be assigned to the appropriate zone and virtual router as well as inbound traffic to the host. In this case, all system services and protocols are allowed for testing. See Example 9-19.

Example 9-19 Configure zone and inbound traffic

```

{primary:node0}[edit]
ibm@J58S-1#
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces reth1.100
set security zones security-zone untrust interfaces reth2.101
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces reth1.200
set security zones security-zone trust interfaces reth2.201

ibm@J58S-1# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
    }
}

```



```

        protocols {
            all;
        }
    }
    interfaces {
        reth1.100;
        reth2.101;
    }
}
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        reth1.200;
        reth2.201;
    }
}

```

Security policy configuration

As shown in “Zone configuration” on page 288, the host inbound traffic is allowed, which can be OSPF, PING, or Traceroute traffic to the router itself. However, in this example, you will test traffic traverse through the network to reach the destination host or IP address. Hence, security policies are required from a security zone to another security zone. Two security policies are defined from untrust to trust zone and vice verse which allows any traffic for testing. See Example 9-20.

Example 9-20 Configure security policy

```

{primary:node0}[edit]
ibm@J58S-1#
set security policies from-zone untrust to-zone trust policy untrust-to-trust
match source-address any
set security policies from-zone untrust to-zone trust policy untrust-to-trust
match destination-address any
set security policies from-zone untrust to-zone trust policy untrust-to-trust
match application any
set security policies from-zone untrust to-zone trust policy untrust-to-trust then
permit

set security policies from-zone trust to-zone untrust policy trust-to-untrust
match source-address any
set security policies from-zone trust to-zone untrust policy trust-to-untrust
match destination-address any
set security policies from-zone trust to-zone untrust policy trust-to-untrust
match application any
set security policies from-zone trust to-zone untrust policy trust-to-untrust then
permit

```

```

{primary:node0}[edit]
ibm@J58S-1# show security policies from-zone untrust to-zone trust
policy untrust-to-trust {
    match {
        source-address any;
        destination-address any;
        application any;
    }
    then {
        permit;
    }
}

{primary:node0}[edit]
ibm@J58S-1# show security policies from-zone trust to-zone untrust
policy trust-to-untrust {
    match {
        source-address any;
        destination-address any;
        application any;
    }
    then {
        permit;
    }
}

```

OSPF configuration

In active/active deployment, dynamic routing protocol is often used to perform dynamic route learning and routing path failover. In this example, you use OSPF protocol, Example 9-21, to form an area 0 network with the rest of the routers.

Example 9-21 Configure OSPF protocol

```

{primary:node0}[edit]
ibm@J58S-1#
set protocols ospf area 0.0.0.0 interface reth1.100
set protocols ospf area 0.0.0.0 interface reth1.200
set protocols ospf area 0.0.0.0 interface reth2.101
set protocols ospf area 0.0.0.0 interface reth2.201
set interfaces lo0 unit 0 family inet address 10.8.8.2/32

{primary:node0}[edit]
ibm@J58S-1# show protocols ospf
area 0.0.0.0 {
    interface reth1.100;
    interface reth1.200;
    interface reth2.101;
    interface reth2.201;
}
ibm@J58S-1# show interfaces lo0
unit 0 {
    family inet {
        address 10.8.8.2/32;
    }
}

```

}

IBM J11M configuration

The IBM Ethernet Router J11M is part of this network layout is complete. Now you must set the following configuration to make sure that the IBM J11M integrates with IBM J58S and IBM J08E. It is assumed that you have some knowledge of this device. For more configuration information, refer to *IBM j-type Ethernet Switches and Routers Implementation*, SG24-7882.

Example 9-22 shows the configuration of physical, IRB, and Loopback interfaces as well as bridge domain and OSPF protocol that are needed in this example.

Example 9-22 Configure IBM J11M

[edit]

ibm@J11M-re0#

Physical Interfaces

```
set interfaces ge-1/0/1 flexible-vlan-tagging
set interfaces ge-1/0/1 encapsulation flexible-ethernet-services
set interfaces ge-1/0/1 unit 100 encapsulation vlan-bridge
set interfaces ge-1/0/1 unit 100 vlan-id 100
set interfaces ge-1/0/1 unit 200 encapsulation vlan-bridge
set interfaces ge-1/0/1 unit 200 vlan-id 200
set interfaces ge-1/0/2 flexible-vlan-tagging
set interfaces ge-1/0/2 encapsulation flexible-ethernet-services
set interfaces ge-1/0/2 unit 101 encapsulation vlan-bridge
set interfaces ge-1/0/2 unit 101 vlan-id 101
set interfaces ge-1/0/2 unit 201 encapsulation vlan-bridge
set interfaces ge-1/0/2 unit 201 vlan-id 201
set interfaces ge-1/0/3 encapsulation ethernet-bridge
set interfaces ge-1/0/3 unit 0 family bridge
set interfaces ge-1/0/4 flexible-vlan-tagging
set interfaces ge-1/0/4 encapsulation flexible-ethernet-services
set interfaces ge-1/0/4 unit 100 encapsulation vlan-bridge
set interfaces ge-1/0/4 unit 100 vlan-id 100
set interfaces ge-1/0/4 unit 101 encapsulation vlan-bridge
set interfaces ge-1/0/4 unit 101 vlan-id 101
set interfaces ge-1/0/4 unit 200 encapsulation vlan-bridge
set interfaces ge-1/0/4 unit 200 vlan-id 200
set interfaces ge-1/0/4 unit 201 encapsulation vlan-bridge
set interfaces ge-1/0/4 unit 201 vlan-id 201
```

IRB and Loopback Interfaces

```
set interfaces irb unit 100 family inet address 192.168.100.3/24 vrrp-group 0
virtual-address 192.168.100.2
set interfaces irb unit 100 family inet address 192.168.100.3/24 vrrp-group 0
priority 200
set interfaces irb unit 100 family inet address 192.168.100.3/24 vrrp-group 0
accept-data
set interfaces irb unit 101 family inet address 192.168.101.3/24 vrrp-group 1
virtual-address 192.168.101.2
set interfaces irb unit 101 family inet address 192.168.101.3/24 vrrp-group 1
priority 200
set interfaces irb unit 101 family inet address 192.168.101.3/24 vrrp-group 1
accept-data
set interfaces lo0 unit 0 family inet address 10.8.8.3/32
```

Bridge Domain

```
set bridge-domains trust-1 vlan-id 200
set bridge-domains trust-1 interface ge-1/0/1.200
set bridge-domains trust-1 interface ge-1/0/4.200
set bridge-domains trust-1 interface ge-1/0/3.0
set bridge-domains trust-2 vlan-id 201
set bridge-domains trust-2 interface ge-1/0/2.201
set bridge-domains trust-2 interface ge-1/0/4.201
set bridge-domains untrust-1 vlan-id 100
set bridge-domains untrust-1 interface ge-1/0/1.100
set bridge-domains untrust-1 interface ge-1/0/4.100
set bridge-domains untrust-1 routing-interface irb.100
set bridge-domains untrust-2 vlan-id 101
set bridge-domains untrust-2 interface ge-1/0/2.101
set bridge-domains untrust-2 interface ge-1/0/4.101
set bridge-domains untrust-2 routing-interface irb.101
```

OSPF Protocol

```
set protocols ospf area 0.0.0.0 interface irb.100
set protocols ospf area 0.0.0.0 interface irb.101
set protocols ospf area 0.0.0.0 interface lo0.0
```

IBM J02M configuration

The IBM Ethernet Router J02M is part of this network layout; therefore, you must set the following configuration to make sure that the IBM J02M integrates with IBM J58S and IBM J08E. It is assumed that you have some knowledge of device. For more configuration information, refer to the *IBM j-type Ethernet Switches and Routers Implementation*, SG24-7882.

Example 9-23 shows the configuration of physical, IRB, and Loopback interfaces as well as bridge domain and OSPF protocol that are needed in this example.

Example 9-23 Configure IBM J02M

[edit]

ibm@J02M#

Physical Interfaces

```
set interfaces ge-1/0/1 flexible-vlan-tagging
set interfaces ge-1/0/1 encapsulation flexible-ethernet-services
set interfaces ge-1/0/1 unit 100 encapsulation vlan-bridge
set interfaces ge-1/0/1 unit 100 vlan-id 100
set interfaces ge-1/0/1 unit 200 encapsulation vlan-bridge
set interfaces ge-1/0/1 unit 200 vlan-id 200
set interfaces ge-1/0/2 flexible-vlan-tagging
set interfaces ge-1/0/2 encapsulation flexible-ethernet-services
set interfaces ge-1/0/2 unit 101 encapsulation vlan-bridge
set interfaces ge-1/0/2 unit 101 vlan-id 101
set interfaces ge-1/0/2 unit 201 encapsulation vlan-bridge
set interfaces ge-1/0/2 unit 201 vlan-id 201
set interfaces ge-1/0/3 encapsulation ethernet-bridge
set interfaces ge-1/0/3 unit 0 family bridge
set interfaces ge-1/0/4 flexible-vlan-tagging
set interfaces ge-1/0/4 encapsulation flexible-ethernet-services
set interfaces ge-1/0/4 unit 100 encapsulation vlan-bridge
```

```

set interfaces ge-1/0/4 unit 100 vlan-id 100
set interfaces ge-1/0/4 unit 101 encapsulation vlan-bridge
set interfaces ge-1/0/4 unit 101 vlan-id 101
set interfaces ge-1/0/4 unit 200 encapsulation vlan-bridge
set interfaces ge-1/0/4 unit 200 vlan-id 200
set interfaces ge-1/0/4 unit 201 encapsulation vlan-bridge
set interfaces ge-1/0/4 unit 201 vlan-id 201

```

IRB and Loopback Interfaces

```

set interfaces irb unit 100 family inet address 192.168.100.4/24 vrrp-group 0
virtual-address 192.168.100.2
set interfaces irb unit 100 family inet address 192.168.100.4/24 vrrp-group 0
priority 200
set interfaces irb unit 100 family inet address 192.168.100.4/24 vrrp-group 0
accept-data
set interfaces irb unit 101 family inet address 192.168.101.4/24 vrrp-group 1
virtual-address 192.168.101.2
set interfaces irb unit 101 family inet address 192.168.101.4/24 vrrp-group 1
priority 200
set interfaces irb unit 101 family inet address 192.168.101.4/24 vrrp-group 1
accept-data
set interfaces lo0 unit 0 family inet address 10.8.8.4/32

```

Bridge Domain

```

set bridge-domains trust-1 vlan-id 200
set bridge-domains trust-1 interface ge-1/0/1.200
set bridge-domains trust-1 interface ge-1/0/4.200
set bridge-domains trust-2 vlan-id 201
set bridge-domains trust-2 interface ge-1/0/2.201
set bridge-domains trust-2 interface ge-1/0/4.201
set bridge-domains trust-2 interface ge-1/0/3.0
set bridge-domains untrust-1 vlan-id 100
set bridge-domains untrust-1 interface ge-1/0/1.100
set bridge-domains untrust-1 interface ge-1/0/4.100
set bridge-domains untrust-1 routing-interface irb.100
set bridge-domains untrust-2 vlan-id 101
set bridge-domains untrust-2 interface ge-1/0/2.101
set bridge-domains untrust-2 interface ge-1/0/4.101
set bridge-domains untrust-2 routing-interface irb.101

```

OSPF Protocol

```

set protocols ospf area 0.0.0.0 interface irb.100
set protocols ospf area 0.0.0.0 interface irb.101
set protocols ospf area 0.0.0.0 interface lo0.0

```

IBM J08E configuration

Since the IBM Ethernet Switch J08E is part of this network layout, you must set the following configuration to make sure that the IBM J08E integrates with all of the other devices. It is assumed that you have some knowledge of the device. For more configuration information, you can refer to *IBM j-type Ethernet Switches and Routers Implementation*, SG24-7882.

Example 9-24 on page 306 shows the configuration of interfaces and OSPF protocol needed in this example.

Example 9-24 Configure IBM J08E

```
{master}[edit]
ibm@J08E-re0#
Interfaces
set interfaces ge-1/0/1 unit 0 family inet address 172.16.1.1/24
set interfaces ge-5/0/0 unit 0 family inet address 192.168.200.2/24
set interfaces ge-5/0/1 unit 0 family inet address 192.168.201.2/24

OSPF Protocol
set protocols ospf area 0.0.0.0 interface ge-5/0/0.0
set protocols ospf area 0.0.0.0 interface ge-5/0/1.0
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface ge-1/0/1.0 passive
```

Verifying the configuration

Here we verify and test the configuration.

Verifying chassis cluster status

To display the failover status of a chassis cluster, use the **show chassis cluster status** command, as shown Example 9-25.

Example 9-25 Verify chassis cluster status

```
{primary:node0}
ibm@J58S-1> show chassis cluster status
Cluster ID: 1
```

Node	Priority	Status	Preempt	Manual failover
Redundancy group: 0 , Failover count: 0				
node0	254	primary	no	no
node1	128	secondary	no	no
Redundancy group: 1 , Failover count: 0				
node0	254	primary	yes	no
node1	128	secondary	yes	no
Redundancy group: 2 , Failover count: 0				
node0	128	secondary	yes	no
node1	254	primary	yes	no

The output shows failover status of the chassis cluster in addition to information about the chassis cluster redundancy groups. As configured, Redundancy group 0 and 1 are primary (active) on node 0 while Redundancy group 2 is primary (active) on node 1.

Verifying chassis cluster interface

To display information about the chassis cluster interfaces, use the **show chassis cluster interfaces** command, as shown Example 9-26 on page 307.

Example 9-26 Verify chassis cluster interface

```
{primary:node0}[edit]
ibm@J58S-1> show chassis cluster interfaces
Control link 0 name: em0
Control link 1 name: em1
```

Redundant-ethernet Information:

Name	Status	Redundancy-group
reth1	Up	1
reth2	Up	2

Interface Monitoring:

Interface	Weight	Status	Redundancy-group
ge-13/0/1	255	Up	1
ge-1/0/1	255	Up	1
ge-13/0/2	255	Up	2
ge-1/0/2	255	Up	2

The output shows the state of the redundant Ethernet interfaces and the status of interfaces the are being monitored.

Verifying chassis cluster statistics

To display information about the chassis cluster services and interfaces, use the **show chassis cluster statistics** command, as shown in Example 9-27.

Example 9-27 Verify chassis cluster statistics

```
{primary:node0}
ibm@J58S-1> show chassis cluster statistics
Control link statistics:
```

```
Control link 0:
  Heartbeat packets sent: 63551
  Heartbeat packets received: 63554
  Heartbeat packet errors: 0
Control link 1:
  Heartbeat packets sent: 0
  Heartbeat packets received: 0
  Heartbeat packet errors: 0
```

Fabric link statistics:

```
Probes sent: 63543
Probes received: 63520
Probe errors: 0
```

Services Synchronized:

Service name	RTOs sent	RTOs received
Translation context	0	0
Incoming NAT	0	0
Resource manager	0	0
Session create	264	107
Session close	183	96
Session change	0	0
Gate create	0	0
Session ageout refresh requests	0	3
Session ageout refresh replies	3	0
IPSec VPN	0	0

Firewall user authentication	0	0
MGCP ALG	0	0
H323 ALG	0	0
SIP ALG	0	0
SCCP ALG	0	0
PPTP ALG	0	0
RPC ALG	0	0
RTSP ALG	0	0
RAS ALG	0	0
MAC address learning	0	0
GPRS GTP	0	0

The output shows the control link statistics (heartbeats sent and received), the fabric link statistics (probes sent and received), and the number of RTOs sent and received for services.

Verifying OSPF neighbor and routes

The following outputs show the OSPF neighbor peering of each device in this example as well as the routes that are learned from OSPF peers. Look for the 172.16.1.0/24 route, which originated from IBM J08E, on the IBM J11M and J02M routing table. You will perform PING and Traceroute using this route. See Example 9-28.

Example 9-28 Verify OSPF neighbor and routes

```
[edit]
ibm@J11M-re0# run show ospf neighbor
```

Address	Interface	State	ID	Pri	Dead
192.168.100.4	irb.100	Full	10.8.8.4	128	33
192.168.100.1	irb.100	Full	10.8.8.2	128	38
192.168.101.4	irb.101	Full	10.8.8.4	128	37
192.168.101.1	irb.101	Full	10.8.8.2	128	37

```

ibm@J02M# run show ospf neighbor
```

Address	Interface	State	ID	Pri	Dead
192.168.100.1	irb.100	Full	10.8.8.2	128	31
192.168.100.3	irb.100	Full	10.8.8.3	128	35
192.168.101.3	irb.101	Full	10.8.8.3	128	39
192.168.101.1	irb.101	Full	10.8.8.2	128	39

```

ibm@J58S-1# run show ospf neighbor
```

Address	Interface	State	ID	Pri	Dead
192.168.100.4	reth1.100	Full	10.8.8.4	128	31
192.168.100.3	reth1.100	Full	10.8.8.3	128	33
192.168.200.2	reth1.200	Full	10.8.8.5	128	36
192.168.101.4	reth2.101	Full	10.8.8.4	128	35
192.168.101.3	reth2.101	Full	10.8.8.3	128	36
192.168.201.2	reth2.201	Full	10.8.8.5	128	32

```

master}[edit]
ibm@J08E-re0# run show ospf neighbor
```

Address	Interface	State	ID	Pri	Dead
192.168.200.1	ge-5/0/0.0	Full	10.8.8.2	128	39
192.168.201.1	ge-5/0/1.0	Full	10.8.8.2	128	38


```
[edit]
ibm@J11M-re0# run show route protocol ospf

inet.0: 15 destinations, 17 routes (15 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.8.8.4/32      *[OSPF/10] 09:49:18, metric 1
                 > to 192.168.100.4 via irb.100
                 to 192.168.101.4 via irb.101
10.8.8.5/32      *[OSPF/10] 09:56:32, metric 2
                 to 192.168.100.1 via irb.100
                 > to 192.168.101.1 via irb.101
172.16.1.0/24    *[OSPF/10] 09:10:34, metric 3
                 > to 192.168.100.1 via irb.100
                 to 192.168.101.1 via irb.101
192.168.200.0/24 [OSPF/10] 10:00:26, metric 2
                 to 192.168.100.1 via irb.100
                 > to 192.168.101.1 via irb.101
192.168.201.0/24 *[OSPF/10] 10:01:03, metric 2
                 to 192.168.100.1 via irb.100
                 > to 192.168.101.1 via irb.101
224.0.0.5/32     *[OSPF/10] 11:41:19, metric 1
                 MultiRecv
```

```
[edit]
ibm@J02M# run show route protocol ospf

inet.0: 22 destinations, 26 routes (22 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.8.8.3/32      *[OSPF/10] 09:49:43, metric 1
                 > to 192.168.100.3 via irb.100
                 to 192.168.101.3 via irb.101
10.8.8.5/32      *[OSPF/10] 09:56:47, metric 2
                 to 192.168.100.1 via irb.100
                 > to 192.168.101.1 via irb.101
172.16.1.0/24    *[OSPF/10] 09:10:49, metric 3
                 to 192.168.100.1 via irb.100
                 > to 192.168.101.1 via irb.101
192.168.200.0/24 [OSPF/10] 09:59:01, metric 2
                 to 192.168.100.1 via irb.100
                 > to 192.168.101.1 via irb.101
192.168.201.0/24 *[OSPF/10] 09:59:01, metric 2
                 to 192.168.100.1 via irb.100
                 > to 192.168.101.1 via irb.101
224.0.0.5/32     *[OSPF/10] 09:59:16, metric 1
                 MultiRecv
```

```
{primary:node0}[edit]
ibm@J58S-1# run show route protocol ospf

inet.0: 21 destinations, 21 routes (21 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```

10.8.8.3/32      *[OSPF/10] 12:14:15, metric 1
                  > to 192.168.100.3 via reth1.100
                  to 192.168.101.3 via reth2.101
10.8.8.4/32      *[OSPF/10] 12:14:04, metric 1
                  > to 192.168.100.4 via reth1.100
                  to 192.168.101.4 via reth2.101
10.8.8.5/32      *[OSPF/10] 12:21:21, metric 1
                  > to 192.168.200.2 via reth1.200
                  to 192.168.201.2 via reth2.201
172.16.1.0/24    *[OSPF/10] 11:35:21, metric 2
                  > to 192.168.200.2 via reth1.200
                  to 192.168.201.2 via reth2.201
224.0.0.5/32     *[OSPF/10] 12:26:01, metric 1
                  MultiRecv

{master}[edit]
ibm@J08E-re0# run show route protocol ospf

inet.0: 29 destinations, 32 routes (29 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.8.8.3/32      *[OSPF/10] 12:15:34, metric 2
                  > to 192.168.200.1 via ge-5/0/0.0
                  to 192.168.201.1 via ge-5/0/1.0
10.8.8.4/32      *[OSPF/10] 12:15:23, metric 2
                  to 192.168.200.1 via ge-5/0/0.0
                  > to 192.168.201.1 via ge-5/0/1.0
192.168.100.0/24 *[OSPF/10] 12:26:25, metric 2
                  > to 192.168.200.1 via ge-5/0/0.0
                  to 192.168.201.1 via ge-5/0/1.0
192.168.101.0/24 *[OSPF/10] 12:26:25, metric 2
                  > to 192.168.200.1 via ge-5/0/0.0
                  to 192.168.201.1 via ge-5/0/1.0
224.0.0.5/32     *[OSPF/10] 12:27:39, metric 1
                  MultiRecv

inet6.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)

```

Testing PING and Traceroute from IBM J11M to IBM J08E

You perform a traceroute from IBM J11M to 172.16.1.1 of IBM J08E. Notice that it is going through 192.168.100.1.

192.168.100.1 is configured on reth1.100 of IBM J58S and is bound to redundancy group 1, which is active on node 0.

Perform a PING test to 172.16.1.1. Example 9-29 shows a successful PING test.

Example 9-29 Perform traceroute and PING test

```

[edit]
ibm@J11M-re0# run traceroute 172.16.1.1
traceroute to 172.16.1.1 (172.16.1.1), 30 hops max, 40 byte packets
 1 192.168.100.1 (192.168.100.1) 1.380 ms 1.240 ms 0.997 ms
 2 172.16.1.1 (172.16.1.1) 1.989 ms 1.416 ms 1.008 ms

```

```
[edit]
ibm@J11M-re0# run ping 172.16.1.1
PING 172.16.1.1 (172.16.1.1): 56 data bytes
64 bytes from 172.16.1.1: icmp_seq=0 ttl=63 time=2.279 ms
64 bytes from 172.16.1.1: icmp_seq=1 ttl=63 time=1.180 ms
64 bytes from 172.16.1.1: icmp_seq=2 ttl=63 time=1.077 ms
64 bytes from 172.16.1.1: icmp_seq=3 ttl=63 time=1.067 ms
^C
--- 172.16.1.1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.067/1.401/2.279/0.509 ms
```

Issue the **show security session flow node 0** command to display the session of the PING (ICMP) test, as shown in Example 9-30.

Example 9-30 View session of PING test

```
{primary:node0}
ibm@J58S-1> show security flow session node 0
node0:
-----

Session ID: 440296029, Policy name: untrust-to-trust/4, State: Active, Timeout: 4
  In: 192.168.100.3/3 --> 172.16.1.1/25626;icmp, If: reth1.100 <-- Ingress packet
  Out: 172.16.1.1/25626 --> 192.168.100.3/3;icmp, If: reth1.200 <-- Egress packet

Session ID: 450592139, Policy name: untrust-to-trust/4, State: Active, Timeout: 2
  In: 192.168.100.3/0 --> 172.16.1.1/25626;icmp, If: reth1.100 <-- Ingress packet
  Out: 172.16.1.1/25626 --> 192.168.100.3/0;icmp, If: reth1.200 <-- Egress packet

Session ID: 450592140, Policy name: untrust-to-trust/4, State: Active, Timeout: 2
  In: 192.168.100.3/1 --> 172.16.1.1/25626;icmp, If: reth1.100 <-- Ingress packet
  Out: 172.16.1.1/25626 --> 192.168.100.3/1;icmp, If: reth1.200 <-- Egress packet
```

The output shows that the ingress packets are received from IBM J11M (192.168.100.3) to the reth1.100 of IBM J58S (node0). Then, egress packets are sent out through reth1.200 of IBM J58S to IBM J08E, as shown in Figure 9-9 on page 312.

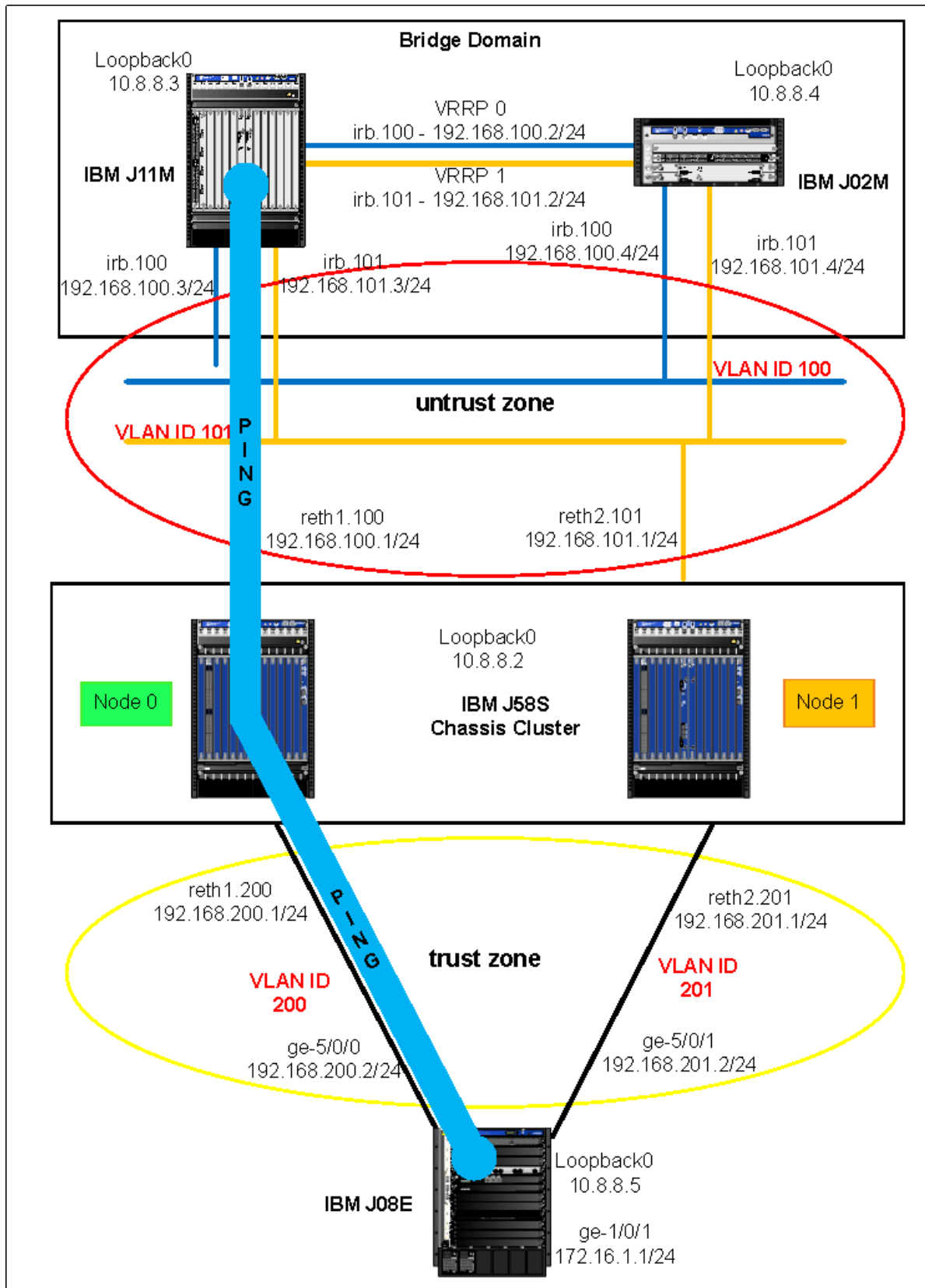


Figure 9-9 Active/active deployment in active traffic path

Testing PING and Traceroute from IBM J02M to IBM J08E

You perform a traceroute from IBM J02M to 172.16.1.1 of IBM J08E. Notice that it is going through 192.168.101.1.

192.168.101.1 is configured on reth2.101 of IBM J58S and is bound to redundancy group 2, which is active on node 1.

Perform a PING test to 172.16.1.1. Example 9-31 shows a successful PING test.

Example 9-31 Perform traceroute and PING test

```
[edit]
ibm@J02M# run traceroute 172.16.1.1
traceroute to 172.16.1.1 (172.16.1.1), 30 hops max, 40 byte packets
 1 192.168.101.1 (192.168.101.1) 0.636 ms 0.562 ms 0.453 ms
 2 172.16.1.1 (172.16.1.1) 1.598 ms 1.343 ms 1.187 ms

[edit]
ibm@J02M# run ping 172.16.1.1
PING 172.16.1.1 (172.16.1.1): 56 data bytes
64 bytes from 172.16.1.1: icmp_seq=0 ttl=63 time=1.742 ms
64 bytes from 172.16.1.1: icmp_seq=1 ttl=63 time=1.390 ms
64 bytes from 172.16.1.1: icmp_seq=2 ttl=63 time=1.434 ms
^C
--- 172.16.1.1 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.390/1.522/1.742/0.157 ms
```

Issue the **show security session flow node 1** command to display the session PING (ICMP) test, as shown in Example 9-32.

Example 9-32 Show session - PING (ICMP) test

```
{primary:node0}[edit]
ibm@J58S-1# run show security flow session node 1
node1:
-----

Session ID: 440167588, Policy name: untrust-to-trust/4, State: Forward, Timeout: 8
  In: 192.168.101.4/0 --> 172.16.1.1/31005;icmp, If: reth2.101 <-- Ingress packet
  Out: 172.16.1.1/31005 --> 192.168.101.4/0;icmp, If: reth1.200 <-- Egress packet

Session ID: 450335246, Policy name: untrust-to-trust/4, State: Forward, Timeout: 8
  In: 192.168.101.4/1 --> 172.16.1.1/31005;icmp, If: reth2.101 <-- Ingress packet
  Out: 172.16.1.1/31005 --> 192.168.101.4/1;icmp, If: reth1.200 <-- Egress packet

Session ID: 450335247, Policy name: untrust-to-trust/4, State: Forward, Timeout: 8
  In: 192.168.101.4/2 --> 172.16.1.1/31005;icmp, If: reth2.101 <-- Ingress packet
  Out: 172.16.1.1/31005 --> 192.168.101.4/2;icmp, If: reth1.200 <-- Egress packet
```

The output shows that the ingress packets are received from IBM J02M (192.168.101.4) to the reth2.101 of IBM J58S (node1). Then, egress packets are sent out through reth1.200 of IBM J58S to IBM J08E, which the traffic flows in Z mode, as shown in Figure 9-10 on page 314.

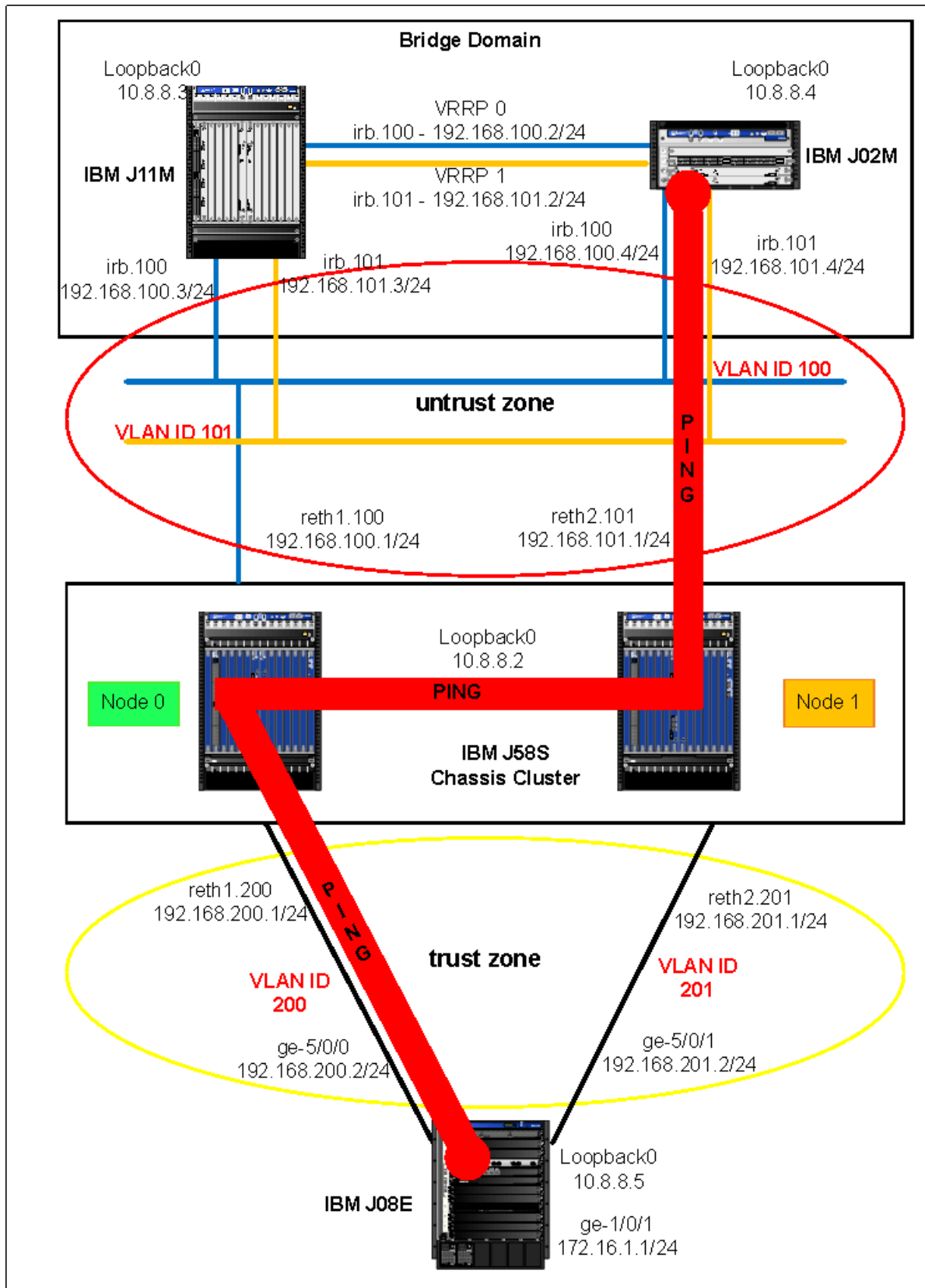


Figure 9-10 Active/active deployment in Z mode traffic path

9.2 Virtual router

This section discusses the features of the virtual router and the techniques of creating multiple virtual routers in an IBM j-type s-series Ethernet Appliance and connect to an IBM j-type m-series Ethernet router.

9.2.1 Virtual router overview

In Junos software, a virtual router is a type of routing instance. A routing instance is a collection of routing tables, interfaces and routing protocol parameters. The set of interfaces belongs to the routing tables and the routing protocol parameters control the information in the routing tables.

To create a virtual router:

1. Create a routing instance with instance type as virtual router.
2. Assign interfaces to a virtual router.
3. Assign interface to a security zone.
4. Define and apply security policy to the virtual router.

Keep the following items in mind when configuring virtual routers:

- ▶ VPN interfaces (st) are currently terminated only in zones that assigned to inet.0 (default routing instance).
- ▶ For self-initiated management traffic (for example system logs and traps), route lookup start with inet.0.
- ▶ Interfaces that are not explicitly members of any virtual router are members of inet.0.

9.2.2 Example: Virtual router deployment

In this section, we configure two virtual routers namely vr-cust1 and vr-cust2 in an IBM J58S. Both virtual routers connect to another virtual router named vr-core in an IBM J11M. vr-cust1 and vr-cust2, each has two interfaces. One interface connects to vr-core and is assigned untrust security zone. Another interface connects to the internal network that resides in trust security zone.

The vr-core serves as a default gateway for vr-cust1 and vr-cust2. Traffic flowing between vr-cust1 and vr-cust2 go through vr-core. This is an example of firewall virtualization deployments where an IBM J58S can be configured to have multiple virtual routers for multiple clients in a Data Center environment with specific client security policies.

Figure 9-11 on page 316 shows the network topology of virtual router deployment.

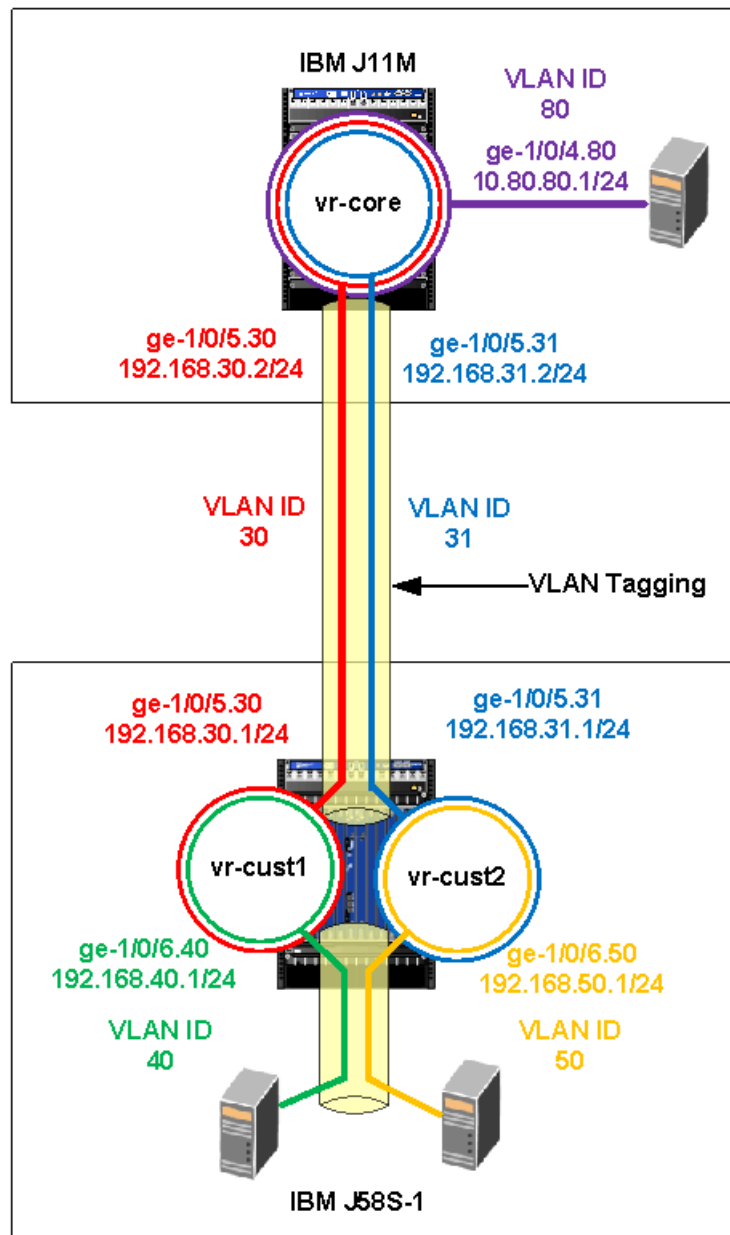


Figure 9-11 Network topology of virtual router

Configuring the interface

First, we assign interface IP addresses to virtual routers. In this example, ge-1/0/5 and ge-1/0/6 are configured as VLAN tagging interfaces. Each interface has two sub-interfaces and are tagged with a VLAN ID.

Example 9-33 on page 317 shows the interfaces configured with IP addresses and assigned a VLAN ID. These interfaces are connected to IBM J11M, which will be assigned in untrust security zone.

Example 9-33 Configure interface ge-1/0/5

```
{primary:node0}[edit]
ibm@J58S-1#
set interfaces ge-1/0/5 vlan-tagging
set interfaces ge-1/0/5 unit 30 vlan-id 30
set interfaces ge-1/0/5 unit 30 family inet address 192.168.30.1/24
set interfaces ge-1/0/5 unit 31 vlan-id 31
set interfaces ge-1/0/5 unit 31 family inet address 192.168.31.1/24

ibm@J58S-1# show interfaces ge-1/0/5
vlan-tagging;
unit 30 {
    vlan-id 30;
    family inet {
        address 192.168.30.1/24;
    }
}
unit 31 {
    vlan-id 31;
    family inet {
        address 192.168.31.1/24;
    }
}
```

Example 9-34 shows the interfaces that are configured with IP addresses and assigned a VLAN ID. These interfaces are connected to an internal network, which will be assigned in the trust security zone.

Example 9-34 Configure interface ge-1/0/6

```
{primary:node0}[edit]
ibm@J58S-1#
set interfaces ge-1/0/6 vlan-tagging
set interfaces ge-1/0/6 unit 40 vlan-id 40
set interfaces ge-1/0/6 unit 40 family inet address 192.168.40.1/24
set interfaces ge-1/0/6 unit 50 vlan-id 50
set interfaces ge-1/0/6 unit 50 family inet address 192.168.50.1/24

ibm@J58S-1# show interfaces ge-1/0/6
vlan-tagging;
unit 40 {
    vlan-id 40;
    family inet {
        address 192.168.40.1/24;
    }
}
unit 50 {
    vlan-id 50;
    family inet {
        address 192.168.50.1/24;
    }
}
```

Configuring the virtual router

Create two routing-instances that are vr-cust1 and vr-cust2. There are several instance types available, and you must select the virtual-router for both vr-cust1 and vr-cust2.

Assign interfaces to vr-cust1 and vr-cust2. As mentioned, vr-cust1 and vr-cust2 has a default gateway pointing to vr-core. Hence, you configure a static default route on each virtual router. See Example 9-35.

Example 9-35 Configure virtual routers

```
{primary:node0}[edit]
ibm@J58S-1#
set routing-instances vr-cust1 instance-type virtual-router
set routing-instances vr-cust1 interface ge-1/0/5.30
set routing-instances vr-cust1 interface ge-1/0/6.40
set routing-instances vr-cust1 routing-options static route 0.0.0.0/0 next-hop
192.168.30.2
set routing-instances vr-cust2 instance-type virtual-router
set routing-instances vr-cust2 interface ge-1/0/5.31
set routing-instances vr-cust2 interface ge-1/0/6.50
set routing-instances vr-cust2 routing-options static route 0.0.0.0/0 next-hop
192.168.31.2

{primary:node0}[edit]
ibm@J58S-1# show routing-instances
vr-cust1 {
    instance-type virtual-router;
    interface ge-1/0/5.30;
    interface ge-1/0/6.40;
    routing-options {
        static {
            route 0.0.0.0/0 next-hop 192.168.30.2;
        }
    }
}
vr-cust2 {
    instance-type virtual-router;
    interface ge-1/0/5.31;
    interface ge-1/0/6.50;
    routing-options {
        static {
            route 0.0.0.0/0 next-hop 192.168.31.2;
        }
    }
}
```

Configuring the zone

In this section, we create security zones and bind interfaces of each virtual router to those security zones.

The vr-cust1 and vr-cust2 has two security zones. First zone (untrust) control traffic from IBM J11M while second zone (trust) restrict traffic from internal network of the virtual router.

Inbound traffic to the virtual router is set to allow all protocols for later test. It is advised to secure the inbound traffic in actual environment. See Example 9-36

Example 9-36 Configuring the security zone

```
{primary:node0}[edit]
ibm@J58S-1#
set security zones security-zone vr-cust1-untrust host-inbound-traffic
system-services all
set security zones security-zone vr-cust1-untrust interfaces ge-1/0/5.30
set security zones security-zone vr-cust1-trust host-inbound-traffic
system-services all
set security zones security-zone vr-cust1-trust interfaces ge-1/0/6.40
set security zones security-zone vr-cust2-untrust host-inbound-traffic
system-services all
set security zones security-zone vr-cust2-untrust interfaces ge-1/0/5.31
set security zones security-zone vr-cust2-trust host-inbound-traffic
system-services all
set security zones security-zone vr-cust2-trust interfaces ge-1/0/6.50

{primary:node0}[edit]
ibm@J58S-1# show security zones
security-zone vr-cust1-untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
    }
    interfaces {
        ge-1/0/5.30;
    }
}
security-zone vr-cust1-trust {
    host-inbound-traffic {
        system-services {
            all;
        }
    }
    interfaces {
        ge-1/0/6.40;
    }
}
security-zone vr-cust2-untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
    }
    interfaces {
        ge-1/0/5.31;
    }
}
security-zone vr-cust2-trust {
    host-inbound-traffic {
        system-services {
            all;
        }
    }
}
```

```

    }
  }
  interfaces {
    ge-1/0/6.50;
  }
}

```

Configuring the security policy

With security zones created, the traffic from and to each virtual router is still not allowed. You must create a security policy from a security zone to another security zone.

As defined in the previous configuration, each virtual router has two security zones. Thus, two security policies are required for each virtual router. The security policies defined in the following Example 9-37 are just policies that allow any IP addresses and protocols to pass through as it is for testing. Again, It is advised to define an appropriate security policy based on security requirement in actual environment.

Example 9-37 Configure security policies

```

{primary:node0}[edit]
ibm@J58S-1#
set security policies from-zone vr-cust1-untrust to-zone vr-cust1-trust policy
vr-cust1-untrust-trust match source-address any
set security policies from-zone vr-cust1-untrust to-zone vr-cust1-trust policy
vr-cust1-untrust-trust match destination-address any
set security policies from-zone vr-cust1-untrust to-zone vr-cust1-trust policy
vr-cust1-untrust-trust match application any
set security policies from-zone vr-cust1-untrust to-zone vr-cust1-trust policy
vr-cust1-untrust-trust then permit
set security policies from-zone vr-cust2-untrust to-zone vr-cust2-trust policy
vr-cust2-untrust-trust match source-address any
set security policies from-zone vr-cust2-untrust to-zone vr-cust2-trust policy
vr-cust2-untrust-trust match destination-address any
set security policies from-zone vr-cust2-untrust to-zone vr-cust2-trust policy
vr-cust2-untrust-trust match application any
set security policies from-zone vr-cust2-untrust to-zone vr-cust2-trust policy
vr-cust2-untrust-trust then permit
set security policies from-zone vr-cust2-trust to-zone vr-cust2-untrust policy
vr-cust2-trust-untrust match source-address any
set security policies from-zone vr-cust2-trust to-zone vr-cust2-untrust policy
vr-cust2-trust-untrust match destination-address any
set security policies from-zone vr-cust2-trust to-zone vr-cust2-untrust policy
vr-cust2-trust-untrust match application any
set security policies from-zone vr-cust2-trust to-zone vr-cust2-untrust policy
vr-cust2-trust-untrust then permit
set security policies from-zone vr-cust1-trust to-zone vr-cust1-untrust policy
vr-cust1-trust-untrust match source-address any
set security policies from-zone vr-cust1-trust to-zone vr-cust1-untrust policy
vr-cust1-trust-untrust match destination-address any
set security policies from-zone vr-cust1-trust to-zone vr-cust1-untrust policy
vr-cust1-trust-untrust match application any
set security policies from-zone vr-cust1-trust to-zone vr-cust1-untrust policy
vr-cust1-trust-untrust then permit

```

```

{primary:node0}[edit]
ibm@J58S-1# show security policies
from-zone vr-cust1-untrust to-zone vr-cust1-trust {
  policy vr-cust1-untrust-trust {
    match {
      source-address any;
      destination-address any;
      application any;
    }
    then {
      permit;
    }
  }
}
from-zone vr-cust2-untrust to-zone vr-cust2-trust {
  policy vr-cust2-untrust-trust {
    match {
      source-address any;
      destination-address any;
      application any;
    }
    then {
      permit;
    }
  }
}
from-zone vr-cust2-trust to-zone vr-cust2-untrust {
  policy vr-cust2-trust-untrust {
    match {
      source-address any;
      destination-address any;
      application any;
    }
    then {
      permit;
    }
  }
}
from-zone vr-cust1-trust to-zone vr-cust1-untrust {
  policy vr-cust1-trust-untrust {
    match {
      source-address any;
      destination-address any;
      application any;
    }
    then {
      permit;
    }
  }
}

```

Configuring the IBM J11M

Since the IBM Ethernet Router J11M is part of this network layout, you must set the following configuration on IBM J11M to make this device work with IBM J58S. It is assumed that you

have some knowledge of device. If you need for more configuration information, refer to *IBM j-type Ethernet Switches and Routers Implementation*, SG24-7882.

IBM J11M has similar features to IBM J58S. A virtual router is referred to in a routing-instance. In this example, IBM J11M has a virtual router called vr-core that connects to vr-cust1 and vr-cust2 of IBM J58S through interface ge-1/0/5 with VLAN tagging. Each sub-interface in ge-1/0/5 is assigned a VLAN ID and an IP address that peers to other virtual router.

The vr-core is configured two static routes pointing to vr-cust1 and vr-cust2 for their internal networks. See Example 9-38.

Example 9-38 Configuring IBM J11M

```
[edit]
ibm@J11M-re0#
set interfaces ge-1/0/5 vlan-tagging
set interfaces ge-1/0/5 unit 30 vlan-id 30
set interfaces ge-1/0/5 unit 30 family inet address 192.168.30.2/24
set interfaces ge-1/0/5 unit 31 vlan-id 31
set interfaces ge-1/0/5 unit 31 family inet address 192.168.31.2/24

set interfaces ge-1/0/4 unit 80 vlan-id 80
set interfaces ge-1/0/4 unit 80 family inet address 10.80.80.1/24

set routing-instances vr-core instance-type virtual-router
set routing-instances vr-core interface ge-1/0/4.80
set routing-instances vr-core interface ge-1/0/5.30
set routing-instances vr-core interface ge-1/0/5.31
set routing-instances vr-core routing-options static route 192.168.40.0/24
next-hop 192.168.30.1
set routing-instances vr-core routing-options static route 192.168.50.0/24
next-hop 192.168.31.1
```

Verifying the routing table of a virtual router

To verify the active route entries in the routing table of a virtual router, use the **show route** command with the find vr option, as shown in Example 9-39.

Example 9-39 Verifying the route entries in the routing table

```
{primary:node0}
ibm@J58S-1> show route | find vr

vr-cust1.inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[Static/5] 00:28:10
                  > to 192.168.30.2 via ge-1/0/5.30
192.168.30.0/24    *[Direct/0] 01:53:56
                  > via ge-1/0/5.30
192.168.30.1/32    *[Local/0] 01:53:56
                  Local via ge-1/0/5.30
192.168.40.0/24    *[Direct/0] 00:59:09
                  > via ge-1/0/6.40
```

```

192.168.40.1/32      *[Local/0] 01:09:30
                    Local via ge-1/0/6.40

vr-cust2.inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0           *[Static/5] 00:28:10
                    > to 192.168.31.2 via ge-1/0/5.31
192.168.31.0/24     *[Direct/0] 01:09:30
                    > via ge-1/0/5.31
192.168.31.1/32     *[Local/0] 01:09:30
                    Local via ge-1/0/5.31
192.168.50.0/24     *[Direct/0] 00:59:09
                    > via ge-1/0/6.50
192.168.50.1/32     *[Local/0] 01:09:30
                    Local via ge-1/0/6.50

```

The output shows the active route entries in vr-cust1 and vr-cust2.

Viewing detailed information of the virtual router

To view detailed information about a specific virtual router, you can use the **show route instance instance-name detail** command, as shown in Example 9-40.

Example 9-40 View detailed information of virtual router

```

{primary:node0}
ibm@J58S-1> show route instance vr-cust1 detail
vr-cust1:
  Router ID: 192.168.30.1
  Type: virtual-router      State: Active
  Interfaces:
    ge-1/0/6.40
    ge-1/0/5.30
  Tables:
    vr-cust1.inet.0          : 5 routes (5 active, 0 holddown, 0 hidden)

{primary:node0}
ibm@J58S-1> show route instance vr-cust2 detail
vr-cust2:
  Router ID: 192.168.31.1
  Type: virtual-router      State: Active
  Interfaces:
    ge-1/0/6.50
    ge-1/0/5.31
  Tables:
    vr-cust2.inet.0          : 5 routes (5 active, 0 holddown, 0 hidden)

```

The output shows the detailed information of a virtual router, which includes router ID, type, interfaces, and tables.

Testing PING between virtual routers

You perform several PING tests from and to each virtual router, as shown in Example 9-41 on page 324.

Example 9-41 Perform PING test

```
{primary:node0}
ibm@J58S-1> ping 192.168.50.1 source 192.168.40.1 routing-instance vr-cust1
PING 192.168.50.1 (192.168.50.1): 56 data bytes
64 bytes from 192.168.50.1: icmp_seq=0 ttl=63 time=2.239 ms
64 bytes from 192.168.50.1: icmp_seq=1 ttl=63 time=4.524 ms
64 bytes from 192.168.50.1: icmp_seq=2 ttl=63 time=3.146 ms
^C
--- 192.168.50.1 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 2.239/3.303/4.524/0.939 ms

{primary:node0}
ibm@J58S-1> ping 10.80.80.1 source 192.168.40.1 routing-instance vr-cust1
PING 10.80.80.1 (10.80.80.1): 56 data bytes
64 bytes from 10.80.80.1: icmp_seq=0 ttl=64 time=1.706 ms
64 bytes from 10.80.80.1: icmp_seq=1 ttl=64 time=2.808 ms
64 bytes from 10.80.80.1: icmp_seq=2 ttl=64 time=2.876 ms
^C
--- 10.80.80.1 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.706/2.463/2.876/0.536 ms

{primary:node0}
ibm@J58S-1> ping 192.168.40.1 source 192.168.50.1 routing-instance vr-cust2
PING 192.168.40.1 (192.168.40.1): 56 data bytes
64 bytes from 192.168.40.1: icmp_seq=0 ttl=63 time=2.447 ms
64 bytes from 192.168.40.1: icmp_seq=1 ttl=63 time=2.791 ms
64 bytes from 192.168.40.1: icmp_seq=2 ttl=63 time=2.890 ms
^C
--- 192.168.40.1 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 2.447/2.709/2.890/0.190 ms

{primary:node0}
ibm@J58S-1> ping 10.80.80.1 source 192.168.50.1 routing-instance vr-cust2
PING 10.80.80.1 (10.80.80.1): 56 data bytes
64 bytes from 10.80.80.1: icmp_seq=0 ttl=64 time=2.195 ms
64 bytes from 10.80.80.1: icmp_seq=1 ttl=64 time=2.615 ms
64 bytes from 10.80.80.1: icmp_seq=2 ttl=64 time=2.393 ms
^C
--- 10.80.80.1 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 2.195/2.401/2.615/0.172 ms
```

The output shows that the PING tests between vr-cust1,vr-cust2 and vr-core are successful.

9.3 More information

For more information, refer to:

- Chassis cluster

<http://www.juniper.net/techpubs/software/junos-security/junos-security10.1/junos-security-swconfig-security/frameset.html>

- Virtual router

http://www.juniper.net/techpubs/en_US/junos10.1/information-products/topic-collections/config-guide-routing/topic-32741.html



Management and monitoring

This chapter provides some of the management and monitoring options that are available on the IBM j-type Ethernet Appliances. In this chapter, we discuss the following topics:

- ▶ Monitoring the device and routing operations
- ▶ Monitoring events and managing system log files
- ▶ Configuring and monitoring alarms

10.1 Configuring SNMP for network management

The Simple Network Management Protocol (SNMP) enables the monitoring of network devices from a central location.

You can use either J-Web Quick Configuration or a configuration editor to configure SNMP.

For more information about SNMP, see the *JUNOS Software Network Management Configuration Guide GA32-0698-02*.

10.1.1 SNMP architecture

Use SNMP to determine where and when a network failure is occurring, and to gather statistics about network performance in order to evaluate the overall health of the network and identify bottlenecks.

Because SNMP is a client/server protocol, SNMP nodes can be classified as either clients, SNMP managers, or servers, SNMP agents. SNMP managers, also called network management systems (NMSs), occupy central points in the network and actively query and collect messages from SNMP agents in the network. SNMP agents are individual processes running on network nodes that gather information for a particular node and transfer the information to SNMP managers as queries are processed. The agent also controls access to the agent's Management Information Base (MIB), the collection of objects that can be viewed or changed by the SNMP manager. Because SNMP agents are individual SNMP processes running on a host, multiple agents can be active on a single network node at any given time.

Communication between the agent and the manager occurs in one of the following forms:

- ▶ **Get, GetBulk, and GetNext** requests: The manager requests information from the agent, and the agent returns the information in a Get response message.
- ▶ **Set** requests: The manager changes the value of a MIB object controlled by the agent, and the agent indicates status in a Set response message.
- ▶ **Traps notification**: The agent sends traps to notify the manager of significant events that occur on the network device.

Management information base

Agents store information in a hierarchical database called the Structure of Management Information (SMI). The SMI resembles a file system. Information is stored in individual files that are hierarchically arranged in the database. The individual files that store the information are known as Management Information Bases (MIBs).

Each MIB contains nodes of information that are stored in a tree structure. Information branches down from a root node to individual leaves in the tree, and the individual leaves comprise the information that is queried by managers for a given MIB. The nodes of information are identified by an object ID (OID). The OID is a dotted integer identifier, 1.3.6.1.2.1.2, for instance) or a subtree name, such as interfaces, that corresponds to an indivisible piece of information in the MIB.

MIBs are either standard or enterprise-specific. Standard MIBs are created by the Internet Engineering Task Force (IETF) and documented in various Rafts. Depending on the vendor, many standard MIBs are delivered with the NMS software. You can also download the standard MIBs from the IETF web site:

<http://www.ietf.org>

For a list of standard and enterprise-specific supported MIBs, see the *JUNOS Software Network Management Configuration Guide GA32-0698-02*.

Enterprise-specific MIBs are developed and supported by a specific equipment manufacturer. If your network contains devices that have enterprise-specific MIBs, you must obtain them from the manufacturer and compile them into your network management software.

To download enterprise MIBs for a device, go to:

http://www.juniper.net/techpubs/software/index_mibs.html

SNMP communities

You can grant access to only specific SNMP managers for particular SNMP agents by creating SNMP communities. The community is assigned a name that is unique on the host. All SNMP requests that are sent to the agent must be configured with the same community name. When multiple agents are configured on a particular host, the community name process ensures that SNMP requests are sorted to only those agents configured to handle the requests.

Additionally, communities allow you to specify one or more addresses or address prefixes to which you want to either allow or deny access. By specifying a list of clients, you can control exactly which SNMP managers have access to a particular agent.

SNMP traps

The **get** and **set** commands that SNMP uses are useful for querying hosts within a network. However, the commands do not provide a means by which events can trigger a notification. For instance, if a link fails, the health of the link is unknown until an SNMP manager next queries that agent.

SNMP traps are unsolicited notifications that are triggered by events on the host. When you configure a trap, you specify the types of events that can trigger trap messages, and you configure a set of targets to receive the generated messages.

SNMP traps enable an agent to notify a network management system (NMS) of significant events. You can configure an event policy action that uses system log messages to initiate traps for events. The traps enable an SNMP trap-based application to be notified when an important event occurs. You can convert any system log message that has no corresponding traps into a trap. This feature helps you to use NMS traps rather than system log messages to monitor the network.

Spoofing SNMP traps

You can use the **request snmp spoof-trap** operational mode command to mimic SNMP trap behavior. The contents of the traps (the values and instances of the objects carried in the trap) can be specified on the command line or they can be spoofed automatically. This feature is useful if you want to trigger SNMP traps and ensure they are processed correctly within your existing network management infrastructure, but find it difficult to simulate the error conditions that trigger many of the traps on the device. For more information, see the *JUNOS Software System Basics and Services Command Reference GA32-0671-02*.

SNMP health monitor

Junos supports RMON as per RFC 2819. RMON can be used to send alerts for MIB variable when they cross Upper and Lower Thresholds. This can be used for various MIB variables which are required. Some good examples are interface stats monitoring, RE CPU monitoring.

RMON alarms and events provide much of the infrastructure needed to reduce the polling overhead from the NMS. However, with this approach, you must set up the NMS to configure specific MIB objects into RMON alarms. This often requires device-specific expertise and customizing of the monitoring application. In addition, some MIB object instances that need monitoring are set only at initialization or change at runtime and cannot be configured in advance. To address these issues, the health monitor extends the RMON alarm infrastructure to provide predefined monitoring for a selected set of object instances (for file system usage, CPU usage, and memory usage) and includes support for unknown or dynamic object instances (such as Junos Software processes).

The SNMP health monitor feature uses existing SNMP remote monitoring (RMON) alarms and traps to monitor a select set of services router characteristics, object instances, like the CPU usage, memory usage, and file system usage. The health monitor feature also monitors the CPU usage of the device's forwarding process, also called a daemon, for example, the chassis process and forwarding process microkernel. You can configure the SNMP health monitor options rising threshold, falling threshold, and interval using the SNMP Quick Configuration page.

A threshold is a test of some SNMP variable against some value, with a report when the threshold value is exceeded. The rising threshold is the upper threshold for a monitored variable. When the current sampled value is greater than or equal to this threshold, and the value at the last sampling interval is less than this threshold, the SNMP health monitor generates an alarm. After the rising alarm, the health monitor cannot generate another alarm until the sampled value falls below the rising threshold and reaches the falling threshold.

The falling threshold is the lower threshold for the monitored variable. When the current sampled value is less than or equal to this threshold, and the value at the last sampling interval is greater than this threshold, the SNMP health monitor generates an alarm. After the falling alarm, the health monitor cannot generate another alarm until the sampled value rises above the falling threshold and reaches the rising threshold.

The interval represents the period of time, in seconds, over which the object instance is sampled and compared with the rising and falling thresholds.

At present, you do not have to configure a separate trap for the SNMP health monitor, because it uses the already existing RMON traps. For more information about RMON events and alarms, see the *JUNOS Software Network Management Configuration Guide GA32-0698-02*.

To display the information collected by the SNMP health monitor, use the following CLI **show snmp health-monitor** commands:

- ▶ **show snmp health-monitor**
- ▶ **show snmp health-monitor alarms**
- ▶ **show snmp health-monitor alarms detail**
- ▶ **show snmp health-monitor logs**

For more information, see the *JUNOS Software System Basics and Services Command Reference GA32-0671-02*.

10.1.2 Before you begin

Before you begin configuring SNMP, complete the following tasks:

1. Establish basic connectivity. See Chapter 5, "Initial configuration" on page 109.
2. Configure network interfaces.

10.1.3 Configuring SNMP with Quick Configuration

J-Web Quick Configuration allows you to define system identification information, create SNMP communities, create SNMP trap groups, and configure health monitor options. Figure 10-1 shows the Quick Configuration page for SNMP.

The screenshot shows the J-Web Quick Configuration page for SNMP. The top navigation bar includes 'Dashboard', 'Configure', 'Monitor', 'Maintain', and 'Troubleshoot'. The 'Configure' tab is selected. The left sidebar shows a tree structure with 'Interfaces', 'Authentication', 'NAT', 'Security', 'IPSec VPN', 'Routing', 'Class of Service', 'System Properties', 'Services', 'DHCP', 'SNMP', and 'CLI Tools'. The 'Services' and 'SNMP' items are highlighted with red circles. The main content area is titled 'Configure' and 'SNMP'. It contains sections for 'Identification' (Contact Information, System Description, Local Engine ID, System Location, System Name Override), 'Communities' (No SNMP communities are defined, Add... button), 'Trap Groups' (No SNMP trap groups are defined, Add... button), and 'Health Monitoring' (Enable Health Monitoring checkbox, Interval, Rising Threshold, Falling Threshold). The 'Apply' button at the bottom is also circled in red.

Figure 10-1 SNMP Quick Configuration

To configure SNMP features with quick configuration:

1. In the J-Web user interface, select **Configure** → **Services** → **SNMP**.
2. Enter information into the Quick Configuration page for SNMP, as described in Table 10-1 on page 332.
3. From the SNMP Quick Configuration page, click **Apply** to apply the configuration and stay on the Quick Configuration page.
4. To check the configuration, see “Verifying the SNMP configuration” on page 339.

Table 10-1 SNMP quick configuration summary

Field	Function	Your action
Identification		
Contact Information	Free-form text string that specifies an administrative contact for the system.	Type any contact information for the administrator of the system, such as name and phone number.
System Description	Free-form text string that specifies a description for the system.	Type any system information that describes the system, for example J56S with 4 PIMs.
Local Engine ID	Provides an administratively unique identifier of an SNMPv3 engine for system identification. The local engine ID contains a prefix and a suffix. The prefix is formatted according to specifications defined in RFC 3411. The suffix is defined by the local engine ID. Generally, the local engine ID suffix is the MAC address of Ethernet management port 0.	Type the MAC address of Ethernet management port 0.
System Location	Free-form text string that specifies the location of the system.	Type any location information for the system, for example room name or rack location.
System Name Override	Free-form text string that overrides the system host name.	Type the name of the system.
Communities		Click Add
Community Name	Specifies the name of the SNMP community.	Type the name of the community being added.
Authorization	Specifies the type of authorization, either read-only or read-write, for the SNMP community being configured.	Select the desired authorization, either read-only or read-write, from the list.
Traps		Click Add .
Trap Group Name	Specifies the name of the SNMP trap group being configured.	Type the name of the SNMP trap group being configured.

Field	Function	Your action
Categories	Specifies which trap categories are added to the trap group being configured.	<ul style="list-style-type: none"> ▶ To generate traps for authentication failures, select Authentication. ▶ To generate traps for chassis and environment notifications, select Chassis. ▶ To generate traps for configuration changes, select Configuration. ▶ To generate traps for link-related notifications, up-down transitions, select Link. ▶ To generate traps for remote operation notifications, select Remote operations. ▶ To generate traps for remote network monitoring (RMON), select RMON alarm. ▶ To generate traps for routing protocol notifications, select Routing. ▶ To generate traps on system warm and cold starts, select Startup. ▶ To generate traps on Virtual Router Redundancy Protocol (VRRP) events, such as new-master or authentication failures, select VRRP events.
Targets	One or more host names or IP addresses that specify the systems to receive SNMP traps generated by the trap group being configured.	<ol style="list-style-type: none"> 1. Enter the host name or IP address, in dotted decimal notation, of the target system to receive the SNMP traps. 2. Click Add.
Health Monitoring		

Field	Function	Your action
Enable Health Monitoring	<p>Enables the SNMP health monitor on the device. The health monitor periodically, the time you specify in the interval field, checks the following key indicators of device health:</p> <ul style="list-style-type: none"> ▶ Percentage of file storage used ▶ Percentage of Routing Engine CPU used ▶ Percentage of Routing Engine memory used ▶ Percentage of memory used for each system process ▶ Percentage of CPU used by the forwarding process ▶ Percentage of memory used for temporary storage by the forwarding process 	<p>Select the check box to enable the health monitor and configure options. If you do not select the check box, the health monitor is disabled.</p> <p>NOTE: If you select only the Enable Health Monitoring check box and do not specify the options, then SNMP health monitoring is enabled with the default values for the options.</p>
Interval	<p>Determines the sampling frequency, in seconds, over which the key health indicators are sampled and compared with the rising and falling thresholds.</p> <p>For example, if you configure the interval as 100 seconds, the values are checked every 100 seconds.</p>	<p>Enter an interval time, in seconds, between 1 and 2147483647.</p> <p>The default value is 300 seconds (5 minutes).</p>
Rising Threshold	<p>Value at which you want SNMP to generate an event, trap and system log message, when the value of a sampled indicator is <i>increasing</i>.</p> <p>For example, if the rising threshold is 90, SNMP generates an event when the value of any key indicator reaches or exceeds 90 percent.</p>	<p>Enter a value between 0 and 100.</p> <p>The default value is 90.</p>
Falling Threshold	<p>Value at which you want SNMP to generate an event, trap and system log message, when the value of a sampled indicator is <i>decreasing</i>.</p> <p>For example, if the falling threshold is 80, SNMP generates an event when the value of any key indicator falls back to 80 percent or less.</p>	<p>Enter a value between 0 and 100.</p> <p>The default value is 80.</p> <p>NOTE: The falling threshold value must be less than the rising threshold value.</p>

10.1.4 Configuring SNMP with a configuration editor

To configure SNMP on a services router, you must perform the following tasks marked (Required):

- ▶ Defining system identification information (Required)
- ▶ Configuring SNMP agents and communities (Required)
- ▶ Managing SNMP trap groups (Required)
- ▶ Controlling access to MIBs (Optional)

Defining system identification information (Required)

Basic system identification information for an appliance can be configured with SNMP and stored in various MIBs. This information can be accessed through SNMP requests and either queried or reset. Table 10-2 identifies types of basic system identification and the MIB object into which each type is stored.

Table 10-2 System identification information and corresponding MIB objects

System information	MIB
Contact	sysContact
System location	sysLocation
System description	sysDescr
System name override	sysName

To configure basic system identification for SNMP:

1. Navigate to the top of the configuration hierarchy in either the J-Web or CLI configuration editor.
2. To configure basic system information using SNMP, perform the configuration tasks described in Table 10-3.
3. If you are finished configuring the network, commit the configuration.
4. To check the configuration, see “Verifying the SNMP configuration” on page 339.

Table 10-3 Configuring basic system identification

Task	J-Web configuration editor	CLI configuration editor
Navigate to the SNMP level in the configuration hierarchy.	1. In the J-Web interface, select Configure → CLI Tools → Point and Click CLI . 2. Next to Snmp, click Configure or Edit . See Figure 10-2 on page 336.	From the [edit] hierarchy level, enter: edit snmp
Configure the system contact information, such as a name and phone number.	In the Contact box, type the contact information as a free-form text string.	Set the contact information: set contact “contact-information”
Configure the system location information, such as a room name and a rack location.	In the Location box, type the location information as a free-form text string.	Set the location information: set location “location-information”
Configure the system description for example, J56S with 4 PIMs.	In the Description box, type the description information as a free-form text string.	Set the description information: set description “description-information”

Task	J-Web configuration editor	CLI configuration editor
Configure a system name to override the system host name defined in the Getting Started Guide for your device.	In the System Name box, type the system name as a free-form text string.	Set the system name: set name <i>name</i>
Configure the local engine ID to use the MAC address of Ethernet management port 0 as the engine ID suffix.	<ol style="list-style-type: none"> 1. Select Engine ID. 2. In the Engine ID choice box, select Use mac address from the list. 3. Click OK. 	Set the engine ID to use the MAC address: set engine-ID use-mac-address

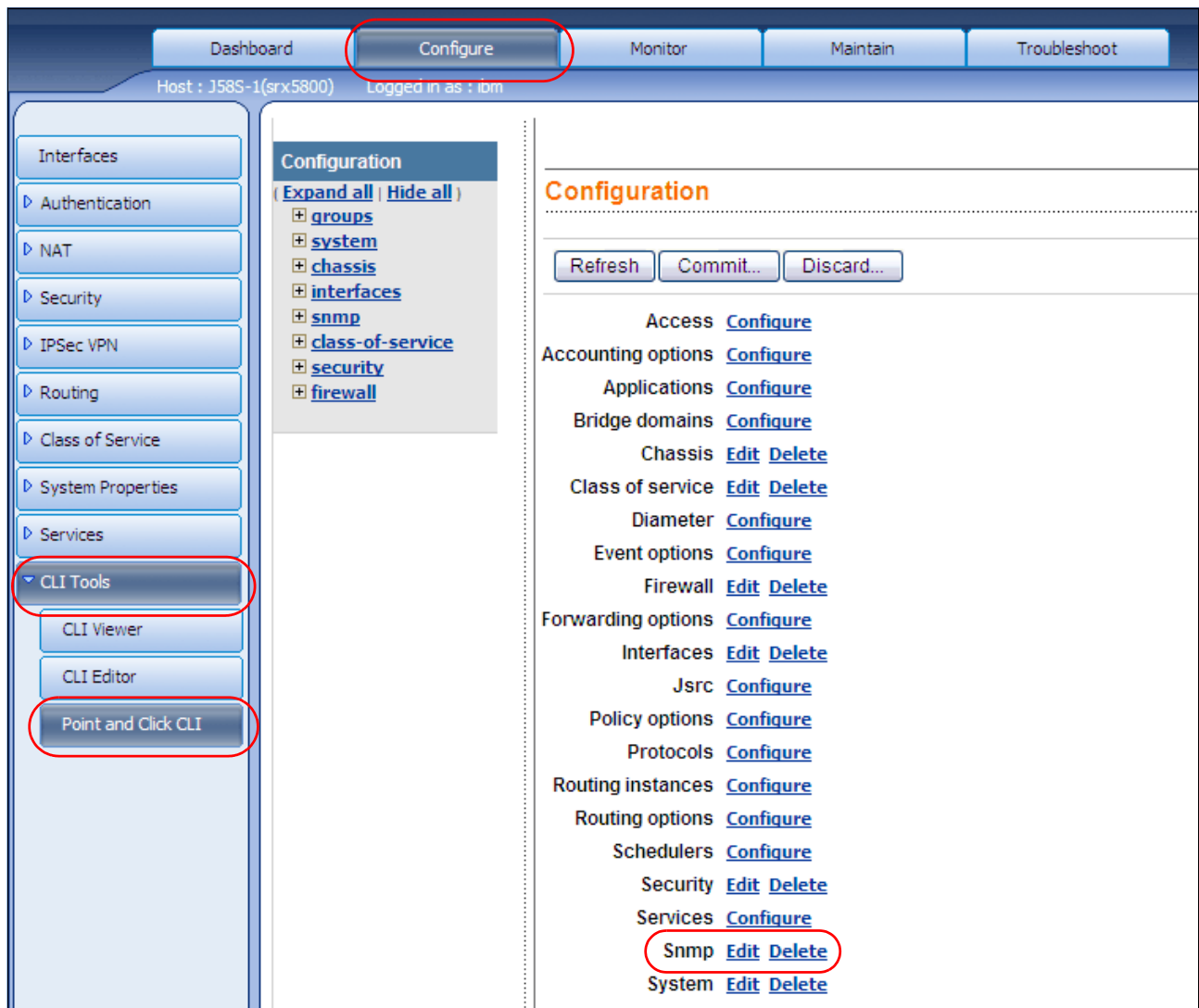


Figure 10-2 J-Web Point and Click CLI

Configuring SNMP agents and communities (Required)

To configure the SNMP agent, you must enable and authorize the network management system access to the services router, by configuring one or more communities. Each community has a community name, an authorization, which determines the kind of access the

network management system has to the device, and, when applicable, a list of valid clients that can access the device.

To configure SNMP communities:

1. Navigate to the top of the configuration hierarchy in either the J-Web or CLI configuration editor.
2. To configure SNMP communities, perform the configuration tasks described in Table 10-4
3. If you are finished configuring the network, commit the configuration.
4. To check the configuration, see “Verifying the SNMP configuration” on page 339.

Table 10-4 Configuring SNMP Agents and Communities

Task	J-Web Configuration Editor	CLI Configuration Editor
Navigate to the SNMP level in the configuration hierarchy.	<ol style="list-style-type: none"> 1. In the J-Web interface, select Configure → CLI Tools → Point and Click CLI. 2. Next to Snmp, click Configure or Edit See Figure 10-2 on page 336	From the [edit] hierarchy level, enter: edit snmp
Create and name a community.	<ol style="list-style-type: none"> 1. Next to Community, click Add new entry. 2. In the Community box, type the name of the community as a free-form text string. 	Create a community: set community community-name
Grant read-write access to the community.	In the Authorization box, select read-write from the list.	Set the authorization to read-write: set community community-name authorization read-write
Allow community access to a client at a particular IP address—for example, at IP address 10.10.10.10	<ol style="list-style-type: none"> 1. Next to Clients, click Add new entry. 2. In the Prefix box, type the IP address, in dotted decimal notation. 3. Click OK. 	Configure client access for the IP address 10.10.10.10: set community community-name clients 10.10.10.10
Allow community access to a group of clients—for example, all addresses within the 10.10.10.0/24 prefix, except those within the 10.10.10.10/29 prefix.	<ol style="list-style-type: none"> 1. Next to Clients, click Add new entry. 2. In the Prefix box, type the IP address prefix 10.10.10.0/24, and click OK. 3. Next to Clients, click Add new entry. 4. In the Prefix box, type the IP address prefix 10.10.10.10/29. 5. Select the Restrict check box. 6. Click OK. 	<ol style="list-style-type: none"> 1. Configure client access for the IP address 10.10.10.0/24: set community community-name clients 10.10.10.0/24 2. Configure client access to restrict the IP addresses 10.10.10.10/29: set community community-name clients 10.10.10.10/29 restrict

Managing SNMP trap groups (Required)

SNMP traps are unsolicited notifications that are generated by conditions on the appliance. When events trigger a trap, a notification is sent to the configured clients for that particular trap group. To manage a trap group, you must create the group, specify the types of traps that are included in the group, and define one or more targets to receive the trap notifications.

To configure SNMP trap groups:

1. Navigate to the top of the configuration hierarchy in either the J-Web or CLI configuration editor.

2. To configure SNMP trap groups, perform the configuration tasks described in Table 10-5.
3. If you are finished configuring the network, commit the configuration.
4. To check the configuration, see “Verifying the SNMP configuration” on page 339.

Table 10-5 Configuring SNMP Trap Groups

Task	J-Web Configuration Editor	CLI Configuration Editor
Navigate to the SNMP level in the configuration hierarchy.	<ol style="list-style-type: none"> 1. In the J-Web interface, select Configure → CLI Tools → Point and Click CLI. 2. Next to Snmp, click Configure or Edit See Figure 10-2 on page 336	From the [edit] hierarchy level, enter: edit snmp
Create a trap group.	<ol style="list-style-type: none"> 1. Next to Trap group, click Add new entry. 2. In the Group name box, type the name of the group as a free-form text string and click OK 	Create a community: set trap-group trap-group-name
Configure the trap group to send all trap notifications to a target IP address—for example, to the IP address 192.174.6.6.	<ol style="list-style-type: none"> 1. Next to Targets, click Add new entry. 2. In the Target box, type the IP address 192.174.6.6, and click OK. 	Set the trap-group target to 192.174.6.6: set trap-group trap-group-name targets 192.174.6.6
Allow community access to a client at a particular IP address—for example, at IP address 10.10.10.10	<ol style="list-style-type: none"> 1. Next to Clients, click Add new entry. 2. In the Prefix box, type the IP address, in dotted decimal notation. 3. Click OK. 	Configure client access for the IP address 10.10.10.10: set community community-name clients 10.10.10.10
Configure the trap group to generate SNMP notifications on authentication failures, environment alarms, and changes in link state for any of the interfaces.	<ol style="list-style-type: none"> 1. Click Categories. 2. Select the Authentication, Chassis, and Link check boxes. 3. Click OK. 	Configure the trap group categories: set trap-group trap-group-name categories authentication chassis link

Controlling access to MIBs (Optional)

By default, an SNMP community is granted access to all MIBs. To control the MIBs to which a particular community has access, configure SNMP views that include the MIBs you want to explicitly grant or deny access to.

To configure SNMP views:

1. Navigate to the top of the configuration hierarchy in either the J-Web or CLI configuration editor.
2. To configure SNMP views, perform the configuration tasks described in Table 10-6 on page 339.
3. If you are finished configuring the network, commit the configuration.
4. To check the configuration, see “Verifying the SNMP configuration” on page 339.

Table 10-6 Configuring SNMP views

Task	J-Web Configuration Editor	CLI Configuration Editor
Navigate to the SNMP level in the configuration hierarchy.	<ol style="list-style-type: none"> 1. In the J-Web interface, select Configure → CLI Tools → Point and Click CLI. 2. Next to Snmp, click Configure or Edit See Figure 10-2 on page 336	From the [edit] hierarchy level, enter: edit snmp
Create a view.	<ol style="list-style-type: none"> 1. Next to Oid, click Add new entry. 2. In the Name box, type the OID of the pingMIB, in either dotted integer or subtree name format. 3. In the View action box, select include from the list, and click OK. 	Set the pingMIB OID value and mark it for inclusion: set view view-name oid 1.3.6.1.2.1.80 include
Configure the view to include a MIB—for example, pingMIB.	<ol style="list-style-type: none"> 1. Next to Targets, click Add new entry. 2. In the Target box, type the IP address 192.174.6.6, and click OK. 	Set the trap-group target to 192.174.6.6: set trap-group trap-group-name targets 192.174.6.6
Configure the view to exclude a MIB—for example, jnxPingMIB.	<ol style="list-style-type: none"> 1. Next to Oid, click Add new entry. 2. In the Name box, type the OID of the jnxPingMIB, in either dotted integer or subtree name format. 3. In the View action box, select exclude from the list, 4. and click OK twice. 	Configure client access for the IP address 10.10.10.10: set community community-name clients 10.10.10.10
Configure the trap group to generate SNMP notifications on authentication failures, environment alarms, and changes in link state for any of the interfaces.	<ol style="list-style-type: none"> 1. Click Categories. 2. Select the Authentication, Chassis, and Link check boxes. 3. Click OK. 	Set the jnxPingMIB OID value and mark it for exclusion: set view view-name oid jnxPingMIB exclude

10.1.5 Verifying the SNMP configuration

To verify the SNMP configuration, perform the following verification task.

Verifying SNMP agent configuration

The following example Example 10-1 shows how to verify that SNMP is running and that requests and traps are being properly transmitted.

Example 10-1 Show SNMP statistics output

```
root@host1> show snmp statistics
```

SNMP statistics:

Input:

```
Packets: 246213, Bad versions: 12 , Bad community names: 12,
Bad community uses: 0, ASN parse errors: 96,
Too bigs: 0, No such names: 0, Bad values: 0,
Read onlys: 0, General errors: 0,
Total request varbinds: 227084, Total set varbinds: 67,
Get requests: 44942, Get nexts: 190371, Set requests: 10712,
Get responses: 0, Traps: 0,
```

```

    Silent drops: 0, Proxy drops: 0, Commit pending drops: 0,
    Throttle drops: 0,
V3 Input:
    Unknown security models: 0, Invalid messages: 0
    Unknown pdu handlers: 0, Unavailable contexts: 0
    Unknown contexts: 0, Unsupported security levels: 1
    Not in time windows: 0, Unknown user names: 0
    Unknown engine ids: 44, Wrong digests: 23, Decryption errors: 0
Output:
    Packets: 246093, Too bigs: 0, No such names: 31561,
    Bad values: 0, General errors: 2,
    Get requests: 0, Get nexts: 0, Set requests: 0,
    Get responses: 246025, Traps: 0

```

The output shows a list of the SNMP statistics, including details about the number and types of packets transmitted. Verify the following information:

- ▶ The number of requests and traps is increasing as expected with the SNMP client configuration.
- ▶ Under Bad community names, the number of bad (invalid) communities is not increasing. A sharp increase in the number of invalid community names generally means that one or more community strings are configured incorrectly.

For a complete description of show snmp statistics output, see the *JUNOS Software System Basics and Services Command Reference GA32-0671-02*.

Verifying SNMP health monitor configuration

The following example Example 10-2 shows how to verify that the SNMP health monitor thresholds are set correctly and that the health monitor is operating properly.

Example 10-2 SNMP health monitor threshold

```

root@host1> show snmp health-monitor
Alarm
Index  Variable description                                Value State
-----
32768  Health Monitor: root file system utilization
      jnxHrStoragePercentUsed.1                        41 active
32769  Health Monitor: /config file system utilization
      jnxHrStoragePercentUsed.2                        0 active
32770  Health Monitor: RE 0 CPU utilization
      jnxOperatingCPU.9.1.0.0                          25 active
32772  Health Monitor: RE 0 memory utilization
      jnxOperatingBuffer.9.1.0.0                       17 active
32774  Health Monitor: Max Kernel Memory Used (%)
      jnxBoxKernelMemoryUsedPercent.0                  2 active

```

The output shows a summary of SNMP health monitor alarms and corresponding log entries:

- ▶ Alarm Index: Alarm identifier
- ▶ Variable description: Object instance being monitored.
- ▶ Value: Current value of the monitored variable in the most recent sample interval.
- ▶ State: Status of the alarm, for example:
 - Active: Entry is fully configured and activated.
 - Falling threshold crossed: Variable value has crossed the lower threshold limit.
 - Rising threshold crossed: Variable value has crossed the upper threshold limit.

Verify that any rising threshold values are greater than the configured rising threshold, and that any falling threshold values are less than the configured falling threshold.

10.2 Monitoring the device and routing operations

This section contains the following topics:

- ▶ Monitoring overview
- ▶ Monitoring interfaces
- ▶ Monitoring events and alarms
- ▶ Monitoring the system
- ▶ Monitoring NAT
- ▶ Monitoring security features

10.2.1 Monitoring overview

Junos software supports a suite of J-Web tools and CLI operational mode commands for monitoring the system health and performance of your device. Monitoring tools and commands display the current state of the device. To use the J-Web interface and CLI operational tools, you must have the appropriate access privileges. For information about configuring access privilege levels, see the *JUNOS Software System Basics and Services Command Reference GA32-0671-02*.

You can use the J-Web Monitor and Manage options to monitor a device. J-Web results are displayed in the browser.

You can also monitor the device with CLI operational mode commands. CLI command output appears on the screen of your console or management device, or you can filter the output to a file. (For complete descriptions of CLI operational mode commands, see the *JUNOS Software CLI Reference GA32-0697-00*, the *JUNOS Software System Basics and Services Command Reference GA32-0671-02*, the *JUNOS Software Interfaces Command Reference GA-32-0672-00*, and the *JUNOS Software Routing Protocols and Policies Command Reference GA32-0673-02*.)

This section contains the following topics:

- ▶ Monitoring terms
- ▶ Filtering command output

Monitoring terms

Before monitoring your device, become familiar with the terms defined in Table 10-7 on page 342.

Table 10-7 Monitoring terms

Term	Definition
Autonomous system (AS)	Network of nodes that route packets based on a shared map of the network topology stored in their local databases.
Internet Control Message Protocol (ICMP)	TCP/IP protocol used to send error and information messages.
Routing table	Database of routes learned from one or more protocols.

Filtering command output

For operational commands that display output, such as the show commands, you can redirect the output into a filter or a file. When you display help about these commands, one of the options listed is |, called a pipe, which allows you to filter the command output.

If you enter the **show configuration** command, the complete device configuration is displayed on the screen. To limit the display to only those lines of the configuration that contain **address**, issue the **show configuration** command using a pipe into the **match** filter as shown in Example 10-3.

Example 10-3 Filter Command output

```
ibm@J58S-1> show configuration | match address
    address 10.1.1.12/24;
    address 10.1.1.13/24;
    address-range low 10.10.10.100 high 10.10.10.200;
    address 10.1.1.1/24;
    address 192.168.1.1/24;
    address 192.168.2.1/24;
    address 192.168.200.2/24;
    address 192.168.201.1/24;
    address 127.0.0.1/32;
    address 192.168.1.1/24;
    address 192.168.2.1/24;
    address-book {
        address out-smtp 10.10.10.100/32;
        address out-pop3 10.10.10.101/32;
        address out-imap 10.10.10.102/32;
        address-set out-mail {
            address out-smtp;
            address out-pop3;
            address out-imap;
        }
    }
    address-book {
        address local-lan 192.168.0.0/16;
        source-address any;
        destination-address any;
        source-address {
```

For a complete list of the filters, type a command, followed by the pipe (|), followed by a question mark (?), shown in Example 10-4.

Example 10-4 Displaying available filters

```
ibm@J58S-1> show configuration | ?
Possible completions:
```

compare	Compare configuration changes with prior version
count	Count occurrences
display	Show additional kinds of information
except	Show only text that does not match a pattern
find	Search for first occurrence of pattern
hold	Hold text without exiting the --More-- prompt
last	Display end of output only
match	Show only text that matches a pattern
no-more	Don't paginate output
request	Make system-level requests
resolve	Resolve IP addresses
save	Save output text to file
trim	Trim specified number of columns from start of line

You can specify complex expressions as an option for the **match** and **except** filters. For more information about command output filtering and creating match expressions, see the *JUNOS Software CLI Reference GA32-0697-00*.

Note: To filter the output of configuration mode commands, use the filter commands provided for the operational mode commands. In configuration mode, an additional filter is supported. See the *JUNOS Software CLI Reference GA32-0697-00*.

10.2.2 Monitoring interfaces

To view general information about all device physical and logical interfaces, select **Monitor** → **Interfaces** in the J-Web user interface, as shown in Figure 10-3.

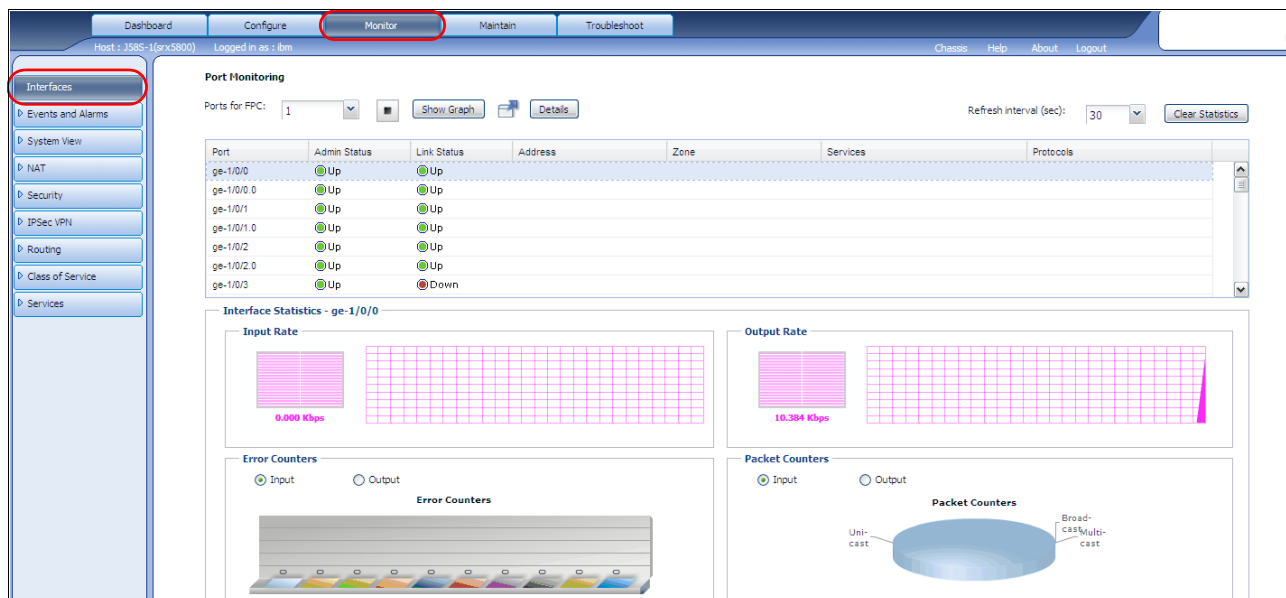


Figure 10-3 Monitor Interfaces J-Web page

Alternatively, you can enter the following show commands in the CLI editor to view interface status and traffic statistics:

```
show interfaces terse
show interfaces detail
```

show interfaces extensive

show interfaces *interface-name*

The J-Web Interfaces page, Figure 10-4, displays the following details about each device interface:

- ▶ Port: Indicates the interface name.
- ▶ Admin Status: Indicates whether the interface is enabled (Up) or disabled (Down).
- ▶ Link Status: Indicates whether the interface is linked (Up) or not linked (Down).
- ▶ Address: Indicates the IP address of the interface.
- ▶ Zone: Indicates whether the zone is an untrust zone or a trust zone.
- ▶ Services: Indicates services that are enabled on the device, such as HTTP and SSH.
- ▶ Protocols: Indicates protocols that are enabled on the device, such as BGP and IGMP.

Port	Admin Status	Link Status	Address	Zone	Services	Protocols
ge-1/0/0	Up	Up				
ge-1/0/0.0	Up	Up				
ge-1/0/1	Up	Up				
ge-1/0/1.0	Up	Up				
ge-1/0/2	Up	Up				
ge-1/0/2.0	Up	Up				
ge-1/0/3	Up	Down				

Figure 10-4 J-Web Interface details

The graph and counters section, shown in Figure 10-5, illustrates the following details:

- ▶ Input Rate graph: Displays interface bandwidth utilization. Input rates are shown in bytes per second.
- ▶ Output Rate graph: Displays interface bandwidth utilization. Output rates are shown in bytes per second.
- ▶ Error Counters chart: Displays input and output error counters in the form of a bar chart.
- ▶ Packet Counters chart: Displays the number of broadcast, unicast, and multicast packet counters in the form of a pie chart. Packet counter charts are supported only for interfaces that support MAC statistics.

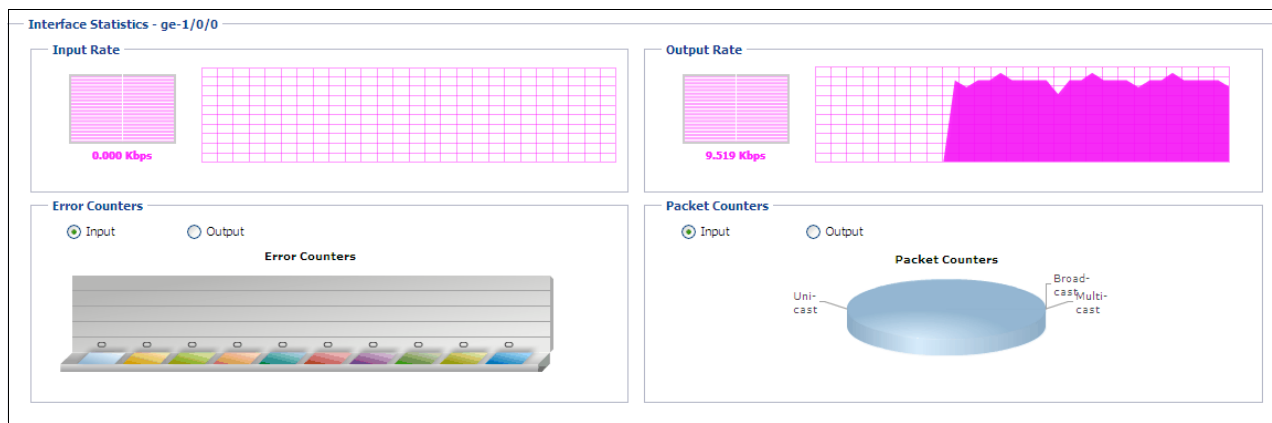


Figure 10-5 J-Web Interface Graphs and Charts

To change the interface display, shown in Figure 10-6 on page 345, use the following options:

- ▶ Port for FPC: Controls the member for which information is displayed.
- ▶ Start/Stop button: Starts or stops monitoring the selected interfaces.
- ▶ Show Graph: Displays input and output packet counters and error counters in the form of charts.

- ▶ Pop-up button: Displays the interface graphs in a separate pop-up window.
- ▶ Details: Displays extensive statistics about the selected interface, including its general status, traffic information, IP address, I/O errors, class-of-service data, and statistics.
- ▶ Refresh Interval: Indicates the duration of time after which you want the data on the page to be refreshed.
- ▶ Clear Statistics: Clears the statistics for the selected interface.



Figure 10-6 J-Web Interface display options

10.2.3 Monitoring events and alarms

For information about monitoring alarms, see 10.4.4, “Checking active alarms” on page 396.

For information about viewing system events, see 10.3.4, “Monitoring system log messages with the J-Web event viewer” on page 386.

10.2.4 Monitoring the system

The system properties include everything from the name and IP address of the device to the resource usage on the Routing Engine.

This topic contains:

- ▶ Monitoring system properties of IBM Ethernet Appliance S-series
- ▶ Monitoring chassis information
- ▶ Monitoring process details

Monitoring system properties of IBM Ethernet Appliance S-series

To view the system properties on an appliance, select Dashboard in the J-Web interface see Figure 10-7 on page 346.

Alternatively, you can view system properties by entering the following **show** commands in the CLI configuration editor:

show system uptime

show system users

show system storage

show version

show chassis hardware

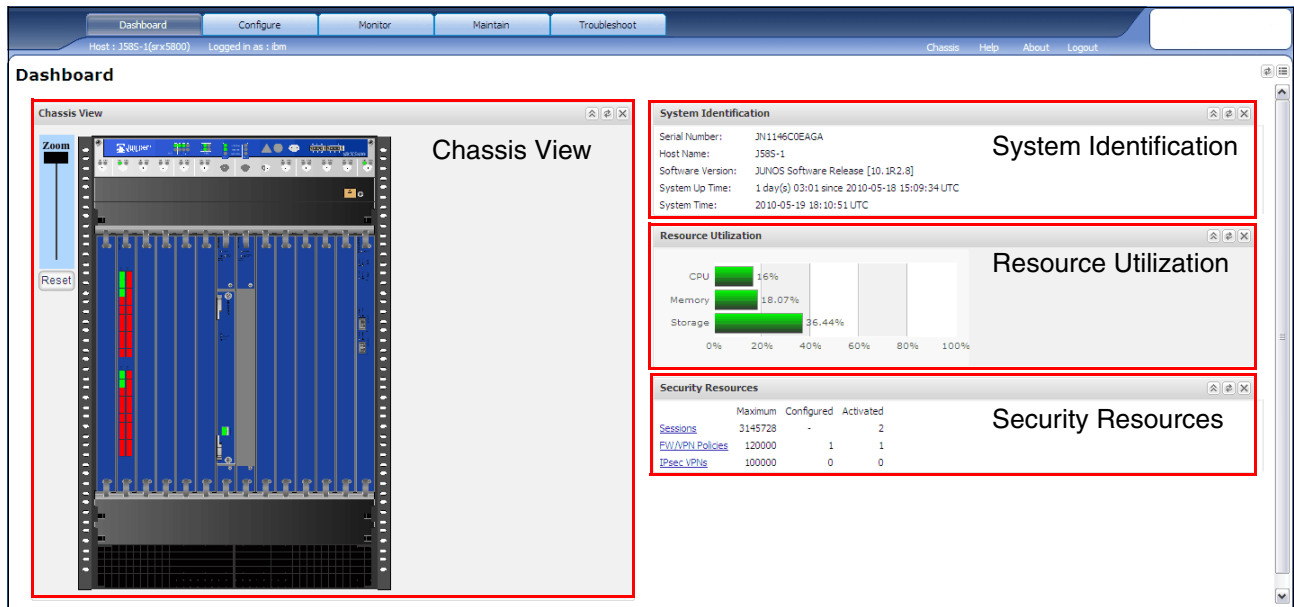


Figure 10-7 J-Web Dashboard page

The Dashboard page, Figure 10-7, displays the following types of information:

- Chassis View (Displayed by default)—Displays an image of the device chassis, including line cards, link states, errors, individual PICs, FPCs, fans, and power supplies.

You can use the Chassis View to link to corresponding configuration and monitoring pages for the device. To link to interface configuration pages for a selected port from the Chassis View, right-click the port in the device image, see Figure 10-8 on page 347 and choose one of the following options:

- Chassis Information—Links to the Chassis page.
- Configure Port: Port-name—**Links to the Configure → Interfaces page for the selected port.**
- Monitor Port: Port-name—**Links to the Monitor → Interfaces page for the selected port.**

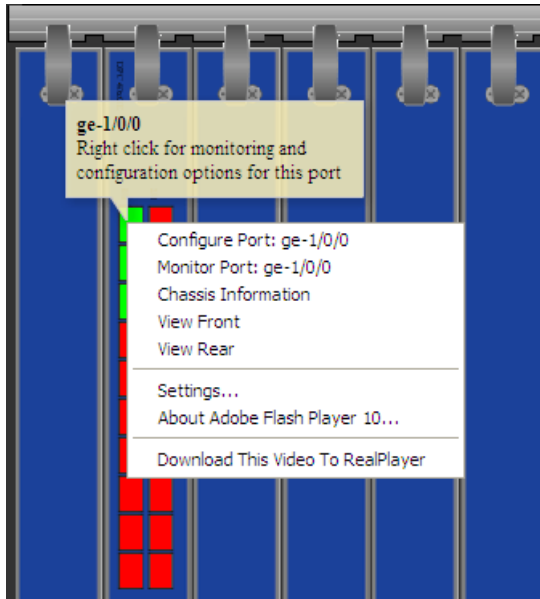


Figure 10-8 J-Web chassis view links

- System Identification (Displayed by default): Displays the device's serial number, host name, current software version, the amount of time since the device was last booted, and the system's time, see Figure 10-9.

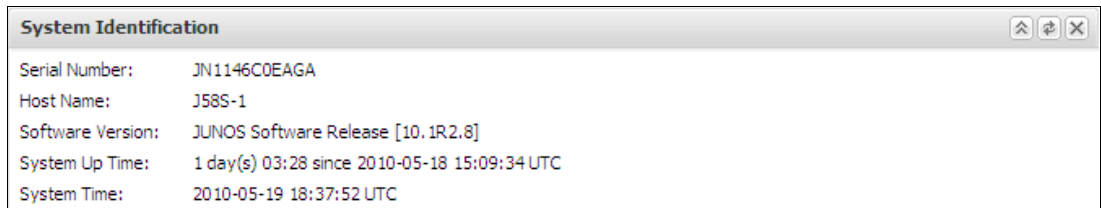


Figure 10-9 J-Web System Identification Panel

Note: The host name that is displayed on this page is defined using the `set system host name` command in the CLI editor. The time zone is defined using the `set system time-zone time-zone` command.

- Resource Utilization (Displayed by default)—Displays the CPU, memory, and storage usage in graph bars, see Figure 10-10.

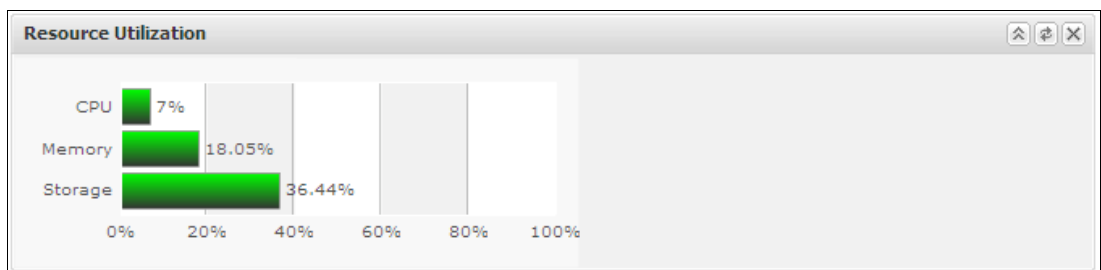


Figure 10-10 J-Web Resource Utilization Panel

- **Security Resources (Displayed by default)**—Displays the current number of sessions running on the device, firewall/VPN policies, and IPsec VPNs security resources details. Click the resource to redirect to details on the Monitor page, see Figure 10-11.



	Maximum	Configured	Activated
Sessions	3145728	-	0
FW/VPN Policies	120000	1	1
IPsec VPNs	100000	0	0

Figure 10-11 J-Web Security Resources Panel

- **Message Logs (Always displayed)**—Displays log messages and errors. You can clear old logs from the Message Logs pane by clicking the Clear button.

In addition to the default views there are a number of other information panels that can be displayed in the Dashboard view:

- **System Alarms (Hidden by default)**: Indicates a missing rescue configuration or software license, where valid. System alarms are preset and cannot be modified.
- **Login Sessions (Hidden by default)**: Displays the log files, temporary files, crash files, and database file details.
- **Chassis Status (Hidden by default)**: Displays the chassis status report in detail.
- **Storage Usage (Hidden by default)**: Displays the storage usage report in detail.
- **File Usage (Hidden by default)**: Displays the file usage of log files, temporary files, crash (core) files, and database files.

These information panels can be selected by clicking the Preferences dialog box in the upper right hand corner of the Dashboard page, see Figure 10-12.

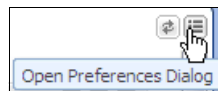


Figure 10-12 Preferences dialog button

When you click the Preferences button a dialog box opens that allows you to select optional information panels and the refresh rate, as shown in Figure 10-13 on page 349.

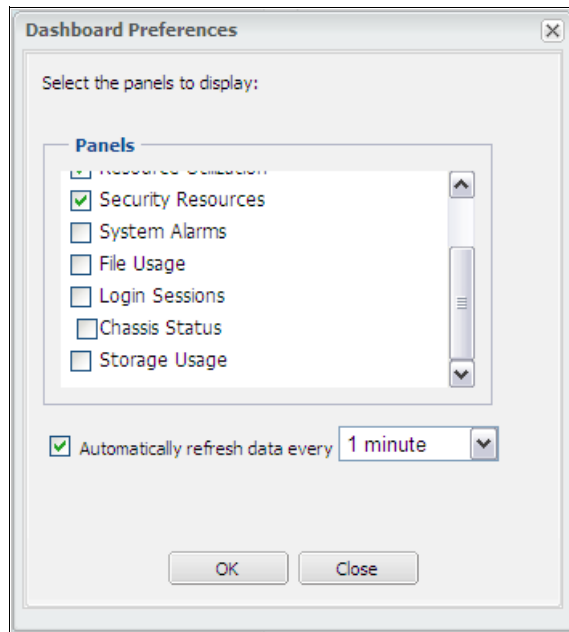


Figure 10-13 Preferences dialog box

Note: To use the Chassis View, a recent version of Adobe® Flash that supports ActionScript and AJAX (Version 9) must be installed. Also note that the Chassis View is displayed by default on the Dashboard page. You can enable or disable it using options in the Dashboard Preference dialog box, but clearing cookies in Internet Explorer also causes the Chassis View to be displayed.

Monitoring chassis information

The chassis properties include the status of hardware components on the device. To view these chassis properties, select **Monitor** → **System View** → **Chassis** Information in the J-Web interface, see Figure 10-14 on page 350.

Alternatively, you can view chassis details by entering the following **show** commands in the CLI configuration editor:

```
show chassis hardware
show chassis routing-engine
show chassis environment
show chassis redundant-power-supply
show redundant-power-supply status
```

Note: Do not install a combination of PIMs in a single chassis that exceeds the maximum power and heat capacity of the chassis. To check PIM power and heat status, use the **show chassis fpc** and **show chassis power-ratings** commands.

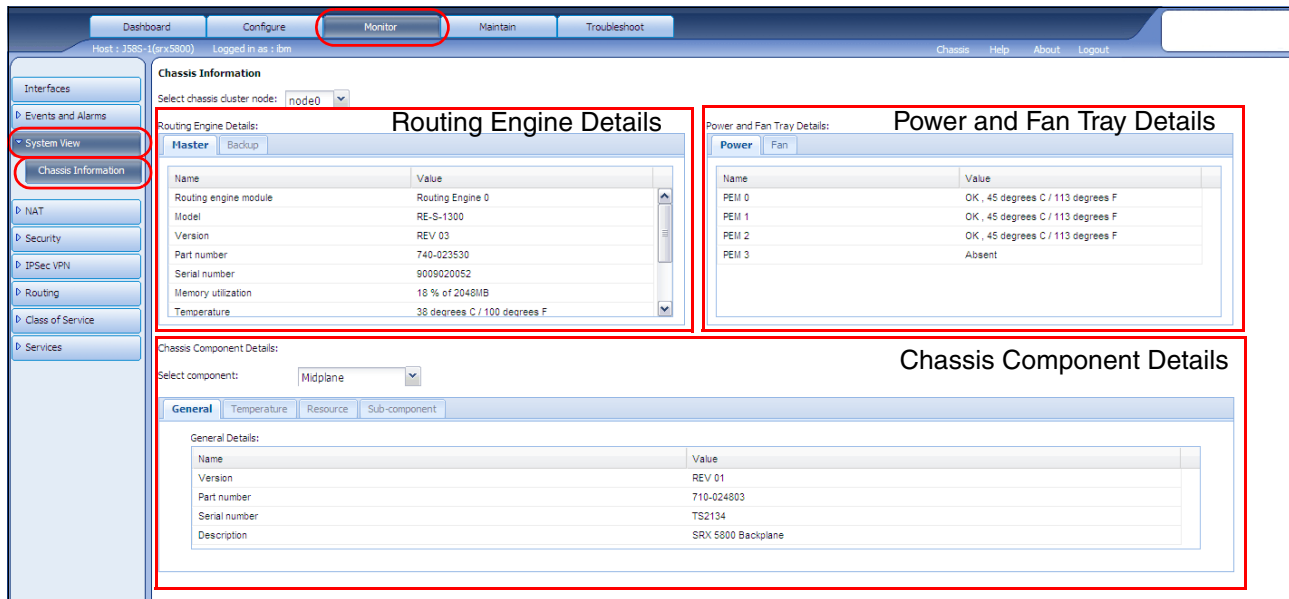


Figure 10-14 J-Web Chassis Information Page

The Chassis Information page, Figure 10-14 displays the following types of information:

- ▶ Routing Engine Details. This section of the page(Figure 10-15) includes the following tabs:
 - Master: The Master tab displays information about the routing engine, including the routing engine module, model number, version, part number, serial number, memory utilization, temperature, and start time. Additionally, this tab displays the CPU load averages for the last 1, 5, and 15 minutes.
 - Backup: If a backup routing engine is available, the Backup tab displays the routing engine module, model number, version, part number, serial number, memory utilization, temperature, and start time. Additionally, this tab displays the CPU load averages for the last 1, 5, and 15 minutes.

Routing Engine Details:	
Master Backup	
Name	Value
Routing engine module	Routing Engine 0
Model	RE-S-1300
Version	REV 03
Part number	740-023530
Serial number	9009020052
Memory utilization	18 % of 2048MB
Temperature	38 degrees C / 100 degrees F

Figure 10-15 J-Web Routing Engine Details

Note: If you need to contact customer support about the device chassis, supply them with the version and serial number displayed in the Routing Engine Details section of the page.

- **Power and Fan Tray Details**—This Details section of the page includes the following tabs:
 - **Power:** The Power tab displays the names of the device's power supply units and their statuses, see Figure 10-16.

Power and Fan Tray Details:

Power Fan	
Name	Value
PEM 0	OK , 45 degrees C / 113 degrees F
PEM 1	OK , 45 degrees C / 113 degrees F
PEM 2	OK , 45 degrees C / 113 degrees F
PEM 3	Absent

Figure 10-16 J-Web Power supply details

- **Fan:** The Fan tab displays the names of the device's fans and their speeds, normal or high. The fan speeds are adjusted automatically according to the current temperature, see Figure 10-17.

Power and Fan Tray Details:

Power Fan	
Name	Value
Top Fan Tray Temp	29 degrees C / 84 degrees F
Top Tray	
Fan 1	Spinning at normal speed
Fan 2	Spinning at normal speed
Fan 3	Spinning at normal speed
Fan 4	Spinning at normal speed
Fan 5	Spinning at normal speed

Figure 10-17 J-Web Fan Tray details

- **Chassis Component Details:** This section of the page includes dropdown list to select components, see Figure 10-18 and also the following tabs.

Chassis Component Details:

Select component: CB 0

General Temperature

General Details:

Name	Version	Part number	Serial number	Description
------	---------	-------------	---------------	-------------

Midplane
FPM Board
PDM
PEM 0
PEM 1
PEM 2
Routing Engine 0
CB 0
CB 1

Figure 10-18 J-Web Chassis Component Selection

- General: The General tab displays the version number, part number, serial number, and description of the selected device component, see Figure 10-19.

Chassis Component Details:

Select component:

General Temperature Resource Sub-component

General Details:

Name	Value
Version	REV 03
Part number	740-023530
Serial number	9009020052
Description	RE-S-1300

Figure 10-19 J-Web Chassis Component General Details

- Temperature: The Temperature tab displays the temperature of the selected device component, if applicable, see Figure 10-20.

Chassis Component Details:

Select component:

General **Temperature** Resource Sub-component

Temperature Details:

Name	Value
	37 degrees C / 98 degrees F
CPU	32 degrees C / 89 degrees F

Figure 10-20 J-Web Chassis Component Temperature Details

- Resource: The Resource tab displays the state, total CPU DRAM, and start time of the selected device component, if applicable, see Figure 10-21.

Chassis Component Details:

Select component:

General Temperature **Resource** Sub-component

Resource Details:

Name	Value
State	master
Total CPU DRAM	2048 MB
Start time	2010-05-18 15:09:34 UTC

Figure 10-21 J-Web Chassis Component Resource Details

Note: On some devices, you can have an FPC state as offline. You may want to put an FPC offline because of an error or if the FPC is not responding. You can put the FPC offline by using the CLI command **request chassis fpc slot number offline**.

- Sub-Component: The Sub-Component tab, Figure 10-22, displays information about the device's sub-components, if applicable. Details include the sub-component's version, part number, serial number, and description.

Chassis Component Details:

Select component: FPC 1

Sub components:				
Name	Version	Part number	Serial number	Description
CPU	REV 02	710-024633	WX3596	SRX5k DPC PMB
PIC 0	...	BULTIN	BULTIN	10x 1GE RichQ
Xcvr 0	REV 01	740-011782	PAR1805	SFP-SX
Xcvr 1	REV 01	740-011613	PCE029W	SFP-SX

Figure 10-22 J-Web Chassis Component Sub-component Details

Monitoring process details

The process details indicates the status of each of the individual processes running on the device. To view these details, select **Monitor** → **System View** → **Process Details** in the J-Web interface, as shown in Figure 10-23.

Dashboard | Configure | Monitor | Maintain | Troubleshoot

Host : J588-1(srv5800) Logged in as : ibm Chassis Help About Logout

Interfaces | Events and Alarms | System View | System Information | Chassis Information | Process Details | NAT | Security | IPsec VPN | Routing | Services

Process Details

CPU Load(1 minute average) 0.0%

Total Memory Utilization 0.0%

PID	Command	State	CPU Load	Memory Utilization	Start Time
29031	jps	runnable	0	0.00 MB	2010-05-26 08:25:39 UTC
29030	php	runnable	0	0.00 MB	2010-05-26 08:25:39 UTC
29029	mgd	sleeping	0	0.00 MB	2010-05-26 08:25:39 UTC
1488	mgd	sleeping	0	0.00 MB	2010-04-26 13:21:13 UTC
1487	cli	sleeping	0	0.00 MB	2010-04-26 13:21:13 UTC
1486	login	sleeping	0	0.00 MB	2010-04-26 13:21:10 UTC
1471	httpd	sleeping	0	0.00 MB	2010-04-26 13:21:04 UTC
1471	httpd	sleeping	0	0.00 MB	2010-04-26 13:21:04 UTC
1471	httpd	sleeping	0	0.00 MB	2010-04-26 13:21:04 UTC
1049	dcd	sleeping	0	0.00 MB	2010-04-26 12:45:53 UTC
1048	snmpd	sleeping	0	0.00 MB	2010-04-26 12:45:53 UTC
1047	mib2d	sleeping	0	0.00 MB	2010-04-26 12:45:53 UTC
1046	pfed	sleeping	0	0.00 MB	2010-04-26 12:45:53 UTC
1045	dfwd	sleeping	0	0.00 MB	2010-04-26 12:45:53 UTC
1044	irsd	sleeping	0	0.00 MB	2010-04-26 12:45:53 UTC
1043	nsd	sleeping	0	0.00 MB	2010-04-26 12:45:53 UTC
1041	sdxcl	sleeping	0	0.00 MB	2010-04-26 12:45:52 UTC

Figure 10-23 J-Web Monitor Process

Alternatively, you can view chassis details by entering the following show commands in the CLI configuration editor:

```
show chassis routing-engine
show system process
```

The Process Details page displays the following types of information for the entire device:

- ▶ CPU Load. Displays the average CPU usage of the device over the last minute in the form of a graph.
- ▶ Total Memory Utilization. Displays the current total memory usage of the device in the form of a graph.

The Process Details page also displays the following types of information for each individual process running on the device:

- ▶ PID: Displays the unique number identifying the process.
- ▶ Value: Displays the name of the process.
- ▶ State: Displays the current state of the process (runnable, sleeping, or unknown).
- ▶ CPU Load: Displays the current CPU usage of the process.
- ▶ Memory Utilization: Displays the current memory usage of the process.
- ▶ Start Time: Displays the time that the process started running.

10.2.5 Monitoring NAT

The J-Web interface provides information about Network Address Translation (NAT).

Monitoring incoming table information

To view Network Address Translation table information, select **Monitor** → **NAT** → **Incoming Table** in the J-Web interface as shown in Figure 10-24.

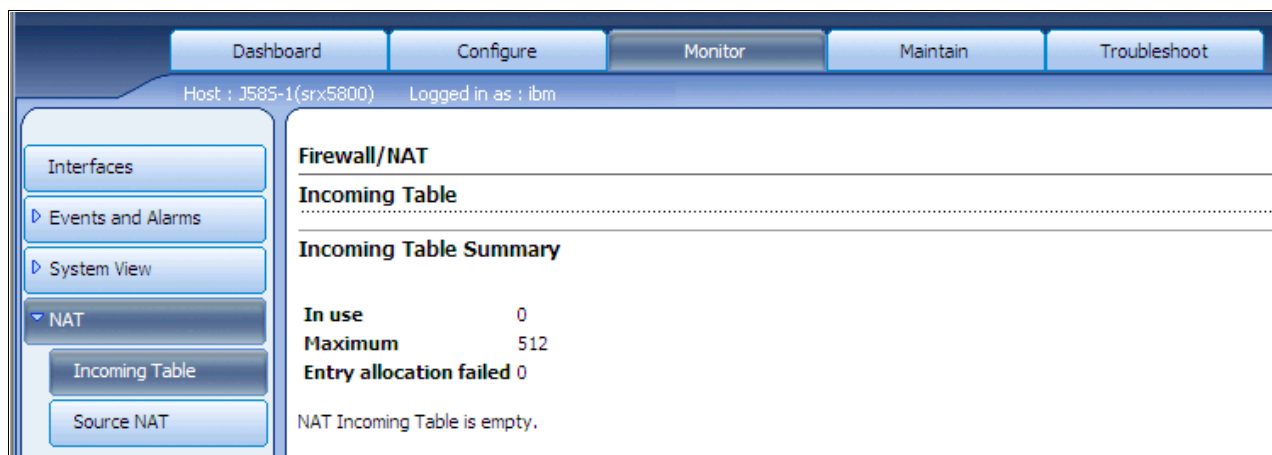


Figure 10-24 Monitoring Incoming Table Information

Alternately, you can enter the following CLI command:

```
show security nat incoming-table
```

Table 10-8 on page 355 summarizes key output fields in the incoming table display.

Table 10-8 Summary of key incoming table output fields

Field	Values	Additional information
Incoming table summary		
In use	Number of entries in the NAT table.	
Maximum	Maximum number of entries possible in the NAT table.	
Entry allocation failed	Number of entries failed for allocation.	
Destination	Destination IP address and port number.	
Host	Host IP address and port number that the destination IP address is mapped to.	
References	Number of sessions referencing the entry.	
Time-out	Time-out, in seconds, of the entry in the NAT table.	
Source-pool	Name of source pool where translation is allocated.	

Monitoring Source NAT information

To view the source Network Address Translation (NAT) summary table and the details of the specified NAT source address pool information, select **Monitor** → **NAT** → **Source NAT** in the J-Web interface, see Figure 10-25.

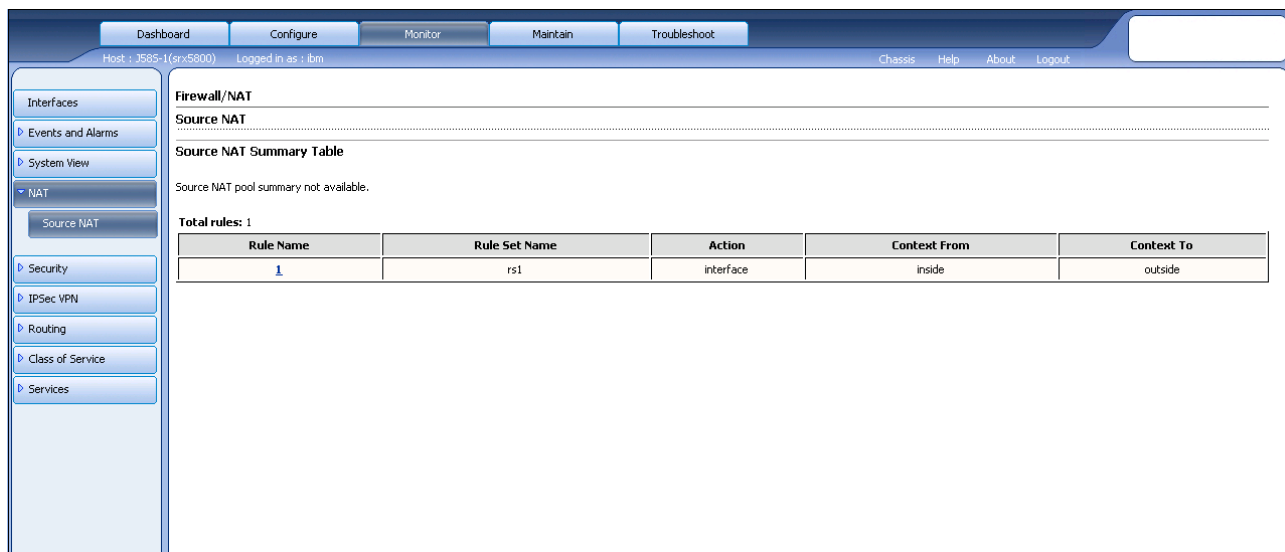


Figure 10-25 J-Web Monitor Source NAT

Alternatively, you can enter the following CLI commands:

```
show security nat source summary
show security nat source pool pool-name
```

Table 10-9 on page 356 summarizes key output fields in the source NAT display.

Table 10-9 Summary of key source NAT output fields

Field	Values	Additional information
Source NAT summary table		
Rule Name	Name of the source pool.	
Rule Set Name	Name of the rule set.	
Action	Actions performed on the NAT rule: <ul style="list-style-type: none"> ▶ Off: Do not perform source NAT. ▶ Pool: Use the specified user-defined address pool to perform source NAT. ▶ Interface: Use the egress interface's IP address to perform source NAT. 	
Context From	Traffic direction source: <ul style="list-style-type: none"> ▶ Interface ▶ Zone ▶ Routing instance 	
Context To	Traffic direction target: <ul style="list-style-type: none"> ▶ Interface ▶ Zone ▶ Routing instance 	

10.2.6 Monitoring security features

The J-Web interface provides the option to monitor security policies, and the Security Screen Counters.

Monitoring policies

Use the monitoring policies feature to view summary information such as names of the source and destination addresses of the policy, name of a preconfigured or custom application defined for the policy, or actions taken on packets matching the policies.

To access policies using J-Web select **Monitor** → **Security** → **Policies** in the J-Web interface. The page layout is as follows and is shown in Figure 10-26 on page 357:

- ▶ Policy list pane: Displays all activated security policies. For details on the policy list, see the Policy List pane
- ▶ Graph pane: Displays the real-time chart for the selected counters. For details on the graph, see the Graph pane.
- ▶ Counters pane: Displays the currently selected policy counters. For details on the counter, see the Counter pane.

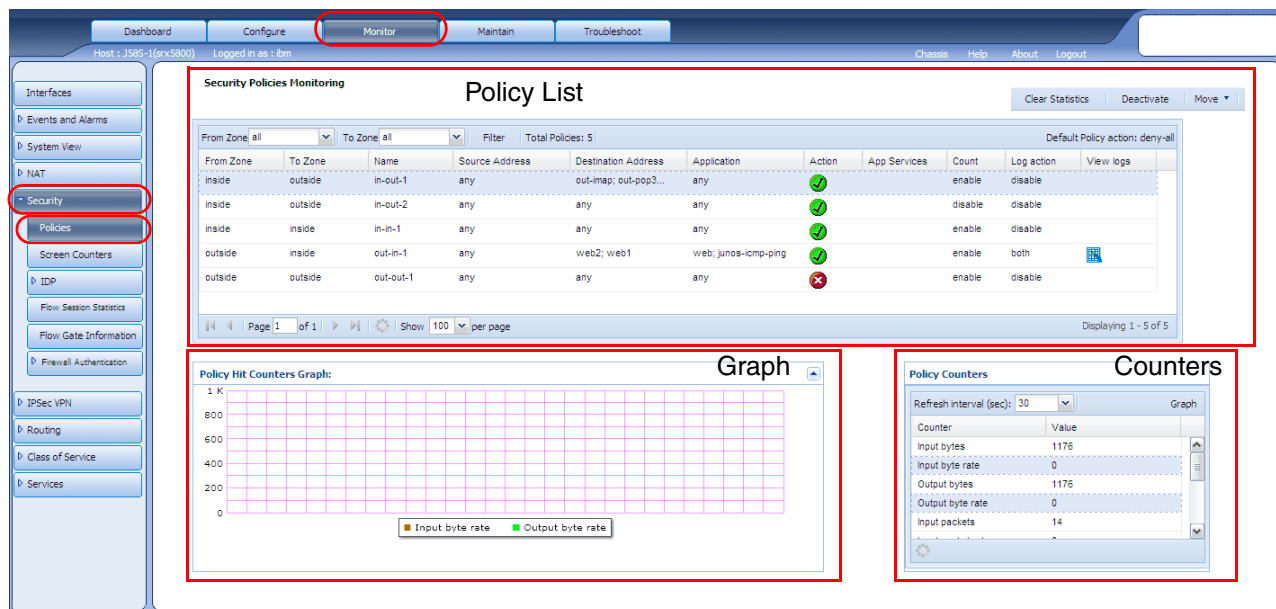


Figure 10-26 J-Web Security Policy Monitoring

Alternatively, you can access policies using the CLI, enter the following CLI commands:

show security policies

show security policies policy-name *policy-name*

Policy list pane

The Policy List pane Figure 10-27 contains a list of policies that are configured on the system for details of the list, see Table 10-10.

- **Clear Statistics:** Clears counters for the selected policies to zero.
- **Deactivate:** Deactivates the policies selected. When you click Deactivate, the commit window pops up and you must confirm the deactivation.
- **Move:** Moves the position of the policy. You have the option to move the policy up, down, top or bottom.

											Clear Statistics	Deactivate	Move
From Zone	To Zone	Name	Source Address	Destination Address	Application	Action	App Services	Count	Log action	View logs			
inside	outside	in-out-1	any	out-imap, out-pop3...	any	✓	enable	1176	disable				
inside	outside	in-out-2	any	any	any	✓	disable	0	disable				
inside	inside	in-in-1	any	any	any	✓	enable	1176	disable				
outside	inside	out-in-1	any	web2; web1	web; junos-icmp-ping	✓	enable	0	both				
outside	outside	out-out-1	any	any	any	✗	enable	14	disable				

Figure 10-27 J-Web Policy List pane

Table 10-10 J-Web Policy List Fields

Field	Values	Additional information
Combo options		
From Zone	Name of the source zone.	

Field	Values	Additional information
To Zone	Name of the destination zone.	
Filter	Filters the policy according to the selected From and To zones and displays only the related policies.	
Total Policies	Number of policies listed in the policy list pane including the default policy.	
Default policy	<p>Actions the device takes on a packet that does not match any user-defined policy:</p> <ul style="list-style-type: none"> ▶ permit-all—Permit all traffic that does not match a policy. ▶ deny-all—Displays the configured default-policy. 	
Policy List Pane		
From Zone	Name of the source zone.	
To Zone	Name of the destination zone.	
Name	Name of the policy.	
Source Address	Names of the source addresses for a policy. Address sets are resolved to their individual names. In this case, only the names are given, not their IP address.	
Destination Address	Name of the destination address (or address set) as it was entered in the destination zone's address book. A packet's destination address must match this value for the policy to apply to it.	
Applications	Name of a preconfigured or custom application whose type the packet matches, as specified at configuration time.	
Action	Status of policy action, either permit or deny/reject	
Application Services	<p>Permitting application services under a policy results in permitting the following possibilities:</p> <ul style="list-style-type: none"> ▶ gprs-gtp-profile: Specify GPRS Tunneling Protocol profile name ▶ idp: Performs Intrusion detection and prevention ▶ redirect-wx: Sets WX redirection ▶ reverse-redirect-wx: Sets WX reverse redirection ▶ uac-policy: Enables unified access control enforcement of policy 	
Count	Enables count for a policy and records the of number of packets hitting the particular policy. For example: the i/p and o/p packets and bytes.	

Field	Values	Additional information
Log	Indicates the log options for log session. The options are: <ul style="list-style-type: none"> ▶ Session initialization ▶ Session close ▶ Both 	
View Logs	Allows access to policy logs	

Graph pane

The graph pane Figure 10-28 appears blank if the counters pane indicates “No data.” If the counters pane contains has data, after the refresh interval, the graph pane begins to draw the graph automatically during the refresh interval for the selected counter

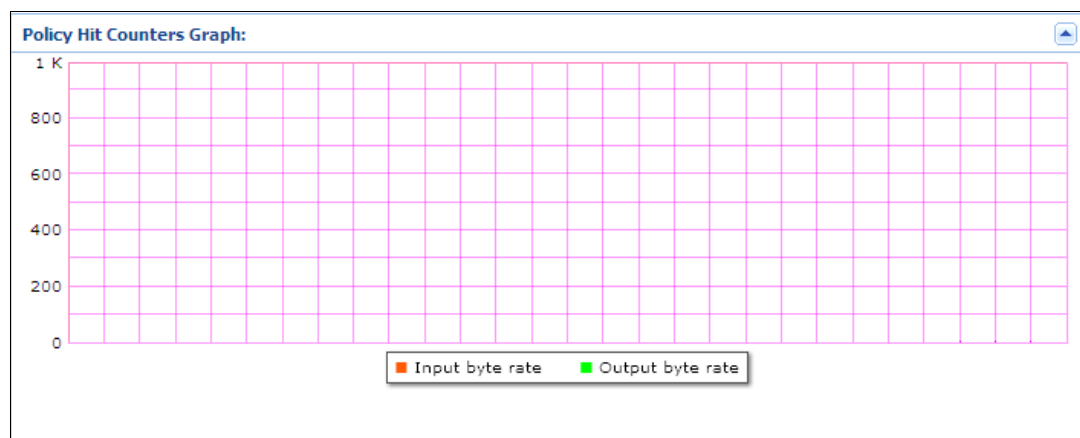


Figure 10-28 J-Web Policy Hit Graph

Policy counter pane

If the selected policy has count enabled, the counters pane Figure 10-29 on page 360 displays counters for that policy. The available counters are:

- ▶ input-bytes
- ▶ input-byte-rate
- ▶ output-bytes
- ▶ output-byte-rate
- ▶ input-packets
- ▶ input-packet-rate
- ▶ output-packets
- ▶ output-packet-rate
- ▶ session-creations
- ▶ session-creation-rate
- ▶ active-sessions

By default, the counters of input-byte-rate and output-byte-rate are selected. The counters pane will be refreshed during the refresh interval.

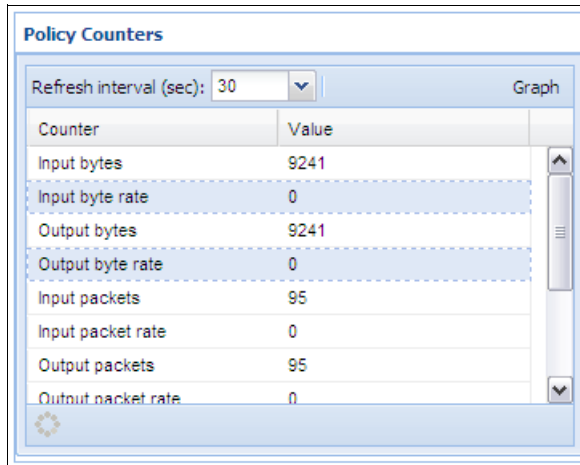


Figure 10-29 J-Web Policy Counters

Monitoring screen counters

To view screen statistics for a specified security zone, select **Monitor** → **Security** → **Screen Counters** in the J-Web interface Figure 10-30, or enter the following CLI command:

show security screen statistics zone zone-name

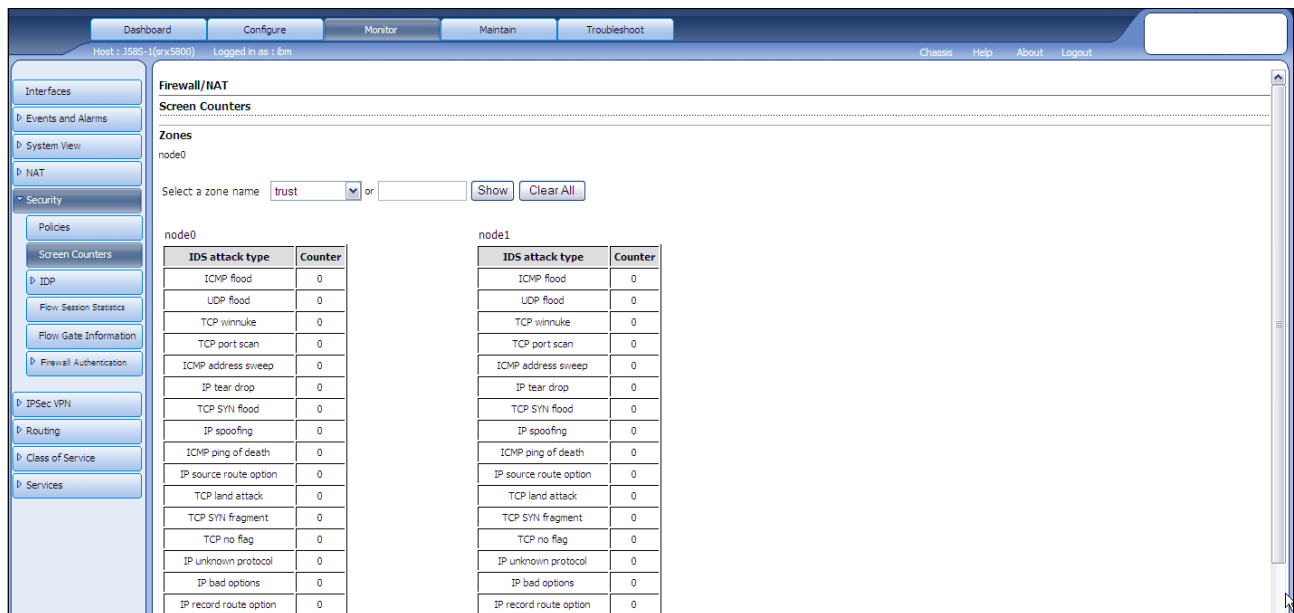


Figure 10-30 J-Web Screen Counters

Table 10-11 summarizes key output fields in the screen counters display.

Table 10-11 Screen counters

Field	Values	Additional Information
Zones		
ICMP Flood	Internet Control Message Protocol (ICMP) flood counter.	An ICMP flood typically occurs when ICMP echo requests use all resources in responding, such that valid network traffic can no longer be processed.
UDP Flood	User Datagram Protocol (UDP) flood counter.	UDP flooding occurs when an attacker sends IP packets containing UDP datagrams with the purpose of slowing down the resources, such that valid connections can no longer be handled.
TCP Winnuke	Number of Transport Control Protocol (TCP) WinNuke attacks.	WinNuke is a denial-of-service (DoS) attack targeting any computer on the Internet running Windows.
TCP Port Scan	Number of TCP port scans.	The purpose of this attack is to scan the available services in the hopes that at least one port will respond, thus identifying a service to target.
ICMP Address Sweep	Number of ICMP address sweeps.	An IP address sweep can occur with the intent of triggering responses from active hosts.
IP Tear Drop	Number of teardrop attacks.	Teardrop attacks exploit the reassembly of fragmented IP packets.
TCP SYN Attack	Number of TCP SYN attacks.	
IP Spoofing	Number of IP spoofs.	IP spoofing occurs when an invalid source address is inserted in the packet header to make the packet appear to come from a trusted source.
ICMP Ping of Death	ICMP ping of death counter.	Ping of death occurs when IP packets are sent that exceed the maximum legal length (65,535 bytes).
IP Source Route	Number of IP source route attacks	
TCP Land Attack	Number of land attacks.	Land attacks occur when attacker sends spoofed SYN packets containing the IP address of the victim as both the destination and source IP address.

Field	Values	Additional Information
TCP SYN Fragment	Number of TCP SYN fragments.	
TCP No Flag	Number of TCP headers without flags set.	A normal TCP segment header has at least one control flag set.
IP Unknown Protocol	Number of unknown Internet protocols.	
IP Bad Options	Number of invalid options.	
IP Record Route Option	Number of packets with the IP record route option enabled.	This option records the IP addresses of the network devices along the path that the IP packet travels.
IP Timestamp Option	Number of IP timestamp option attacks.	This option records the time (in Universal Time) when each network device receives the packet during its trip from the point of origin to its destination.
IP Security Option	Number of IP security option attacks.	
IP Loose route Option	Number of IP loose route option attacks.	This option specifies a partial route list for a packet to take on its journey from source to destination.
IP Strict Source Route Option	Number of IP strict source route option attacks.	This option specifies the complete route list for a packet to take on its journey from source to destination.
IP Stream Option	Number of stream option attacks.	This option provides a way for the 16-bit SATNET stream identifier to be carried through networks that do not support streams.
ICMP Fragment	Number of ICMP fragments.	Because ICMP packets contain very short messages, there is no legitimate reason for ICMP packets to be fragmented. If an ICMP packet is so large that it must be fragmented, something is amiss.
ICMP Large Packet	Number of large ICMP packets.	
TCP SYN FIN Packet	Number of TCP SYN FIN packets.	
TCP FIN without ACK	Number of TCP FIN flags without the acknowledge (ACK) flag.	

Field	Values	Additional Information
TCP SYN-ACK-ACK Proxy	Number of TCP flags enabled with SYN-ACK-ACK.	To prevent flooding with SYN-ACK-ACK sessions, you can enable the SYN-ACK-ACK proxy protection screen option. After the number of connections from the same IP address reaches the SYN-ACK-ACK proxy threshold, Junos Software rejects further connection requests from that IP address.
IP Block Fragment	Number of IP block fragments.	

10.2.7 Monitoring IDP

Here we discuss about Monitoring IDP Status. IDP monitoring pages allow you to display detailed information about the IDP Status, Memory, Counters, Policy rulebase statistics and Attack table statistics.

Monitoring IDP status

To view Intrusion Detection and Prevention (IDP) table information, select **Monitor** → **Security** → **IDP** → **Status** in the J-Web interface Figure 10-31, or enter the following CLI commands:

```
show security idp status
```

```
show security idp memory
```

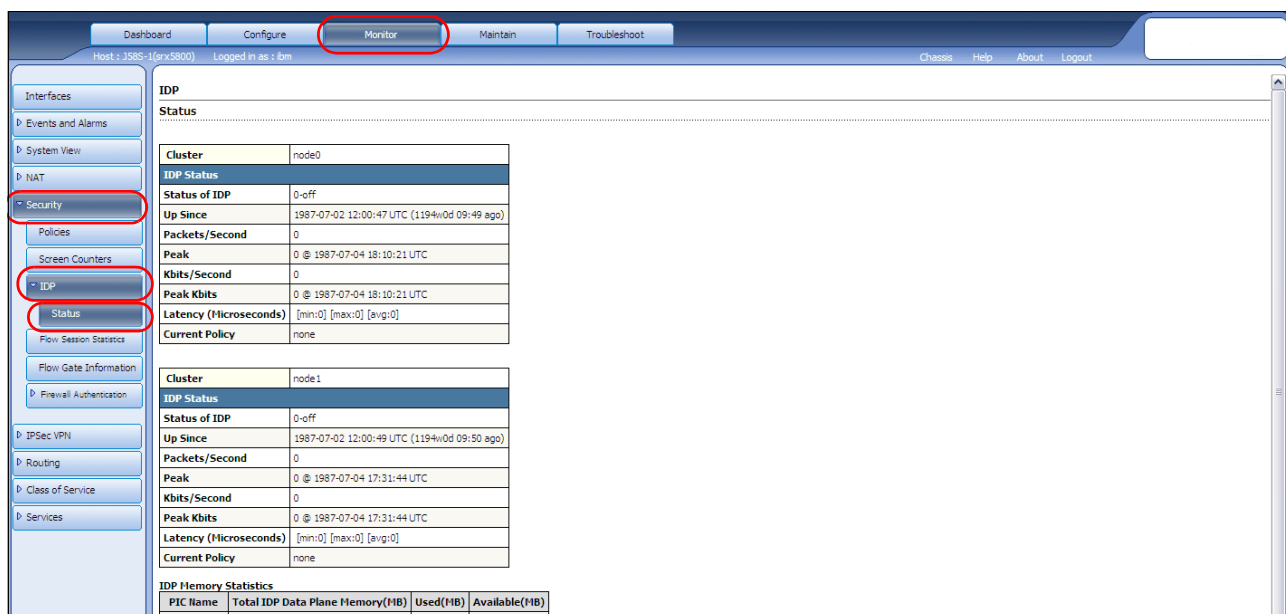


Figure 10-31 J-Web IDP Status

Table 10-12 on page 364 summarizes key output fields in the IDP display.

Table 10-12 J-Web Policy List Fields

Field	Values	Additional information
IDP status		
Status of IDP	Displays the status of the current IDP policy.	
Up Since	Displays the time from when the IDP policy first began running on the system.	
Packets/Second	Displays the number of packets received and returned per second.	
Peak	Displays the maximum number of packets received per second and the time when the maximum was reached.	
Kbits/Second	Displays the aggregated throughput (kilobits per second) for the system.	
Peak Kbits	Displays the maximum kilobits per second and the time when the maximum was reached.	
Latency Microseconds	Displays the delay, in microseconds, for a packet to receive and return by a node.	
Current Policy	Displays the name of the current installed IDP policy.	
IDP Memory Statistics	Displays the status of all IDP data plane memory.	
PIC Name	Displays the name of the PIC.	
Total IDP Data Plane Memory (MB)	Displays the total memory space, in megabytes, allocated for the IDP data plane.	
Used (MB)	Displays the used memory space, in megabytes, for the data plane.	
Available (MB)	Displays the available memory space, in megabytes, for the data plane.	

10.2.8 Monitoring flow session statistics

The J-Web interface provides session statistics according to the session filter you select on the Flow Session Statistics page.

This section contains the following topics:

- ▶ Monitoring Flow Session Statistics Summary Information
- ▶ Monitoring Flow Information for All Sessions
- ▶ Monitoring Flow Information for Application Sessions
- ▶ Monitoring Flow Session Destination Port Information
- ▶ Monitoring Flow Session Destination Prefix Information
- ▶ Monitoring Flow Session Interface Information
- ▶ Monitoring Flow Session Protocol Information
- ▶ Monitoring Flow Session Resource Manager
- ▶ Monitoring Flow Session Identifier Session
- ▶ Monitoring Flow Session Source Port Information

- ▶ Monitoring Flow Session Source Prefix Information
- ▶ Monitoring Flow Session Tunnel Information

Monitoring flow session statistics summary information

To view summary information about existing sessions, including types of sessions, active and failed sessions, and the maximum allowed number of sessions, select

Monitor → **Security** → **Flow Session Statistics** in the J-Web interface, see Figure 10-32.

Then select **summary** from the Session Filter list and click **Show**. Alternatively, enter the following CLI command:

```
show security flow session summary
```

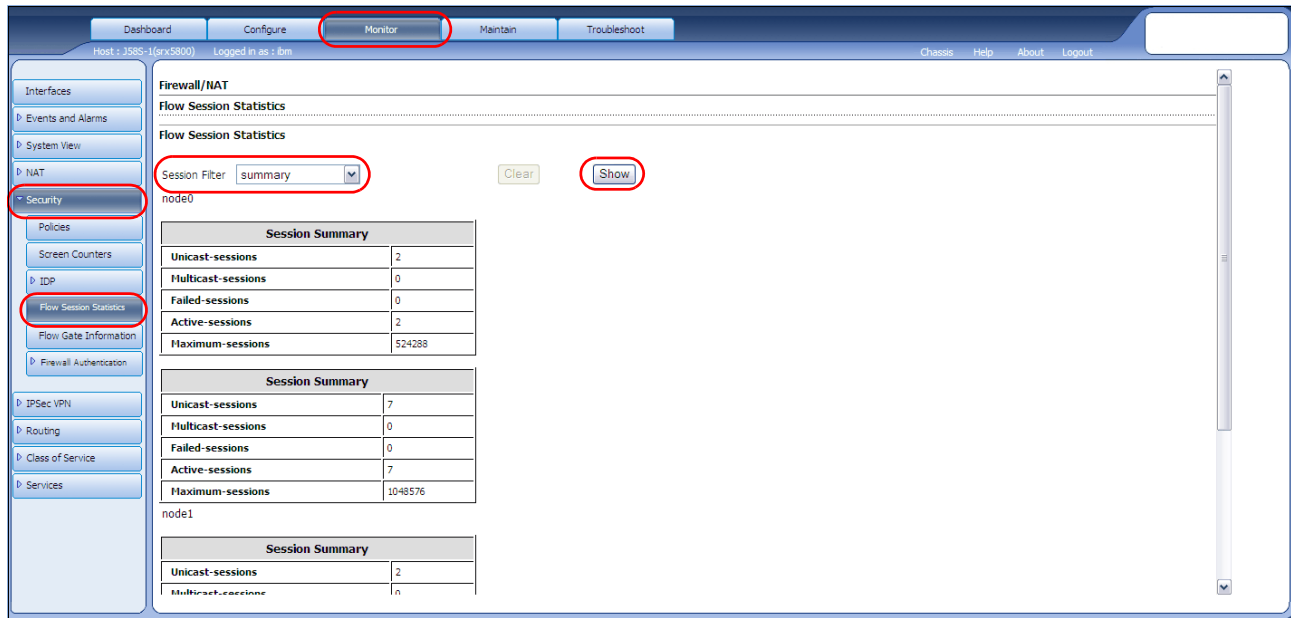


Figure 10-32 J-Web Flow Session Monitoring display

Table 10-13 summarizes key output fields in the flow session statistics display.

Table 10-13 Summary of Key Flow Session Statistics Output Fields

Field	Values	Additional information
Flow Session Statistics: session filter—summary (By default)		
Unicast-sessions	Total number of active unicast sessions.	
Multicast-sessions	Total number of active multicast sessions.	
Failed-sessions	Total number of failed sessions.	
Active-sessions	Total number of active sessions.	
Maximum-sessions	Maximum number of supported sessions.	

Monitoring flow information for all sessions

To view information about all currently active security sessions on the device, select **Monitor** → **Security** → **Flow Session Statistics** in the J-Web interface, see Figure 10-32 on page 365. Then select **all** from the Session Filter list and click **Show**. To view information about the incoming and outgoing source and destination addresses and the protocol and interface for a specific session, select the session ID on the Flow Session Statistics page. Alternatively, enter the following CLI command:

```
show security flow session
```

You can also monitor nodes using the command `show security flow session node node-id` when using HA feature. Here the *node-id* is the Identification number of the node and it can be 0 or 1. Example 10-5 shows a sample output of the Flow Session statistics for node0 and node1

Example 10-5 Flow Session Statistics for node0 and node 1

```
node0:
-----
Session ID: 2, Policy name: self-traffic-policy/1, State: Active, Timeout:
1800
In: 172.24.241.53/50045 --> 172.19.101.34/22;tcp, If: ge-0/0/0.0
Out: 172.19.101.34/22 --> 172.24.241.53/50045;tcp, If: .local..0
1 sessions displayed

node1:
-----
Session ID: 2, Policy name: self-traffic-policy/1, State: Backup, Timeout:
14400
In: 172.24.241.53/50045 --> 172.19.101.34/22;tcp, If: ge-0/0/0.0
Out: 172.19.101.34/22 --> 172.24.241.53/50045;tcp, If: .local..0
1 sessions displayed
```

Session ID can be used to get more details on each session using the command `show security flow session session-identifier`

Table 10-14 summarizes key output fields in the flow all session display.

Table 10-14 Summary of Key Flow All Session Information Output Fields

Field	Values	Additional information
Flow Session Statistics: session filter—all		
Session ID	Number that identifies the session. Use this ID to get more information about the session.	
Policy name	Policy that permitted the traffic.	
Timeout	Idle timeout after which the session expires.	
Flow Session Statistics: Session ID		
In	Incoming flow (source and destination IP addresses, application protocol, and interface).	

Field	Values	Additional information
Out	Reverse flow (source and destination IP addresses, application protocol, and interface).	
Flow Session Statistics: State		
Active	Active state refers to Active session on Primary node.	
Backup	Backup state refers to Backup session on Secondary/passive Node.	

Monitoring flow information for application sessions

To view information about each session of the specified application type, select **Monitor** → **Security** → **Flow Session Statistics** in the J-Web interface, see Figure 10-32 on page 365. Then select **application** from the Session Filter list and click **Show**. Alternatively, enter the following CLI command:

```
show security flow session application application-name
```

Table 10-15 summarizes key output fields in the flow session application display.

Table 10-15 Summary of key flow application session information output fields

Field	Values	Additional information
Flow session statistics: session filter—application		
Session ID	Number that identifies the session. Use this ID to get more information about the session.	
Policy name	Policy that permitted the traffic.	
Timeout	Idle timeout after which the session expires.	
In	Incoming flow (source and destination IP addresses, application protocol, and interface).	
Out	Reverse flow (source and destination IP addresses, application protocol, and interface).	

Monitoring flow session destination port information

To view information about each session that uses the specified destination port, select **Monitor** → **Security** → **Flow Session Statistics** in the J-Web interface, see Figure 10-32 on page 365. Then select **destination port** from the Session Filter list and click **Show**. Alternatively, enter the following CLI command:

```
show security flow session destination-port destination-port-number
```

Table 10-16 summarizes key output fields in the flow session destination port display.

Table 10-16 Summary of key flow destination port session information output fields

Field	Values	Additional information
Flow session statistics: session filter—destination port		
Session ID	Number that identifies the session. Use this ID to get more information about the session.	
Policy name	Policy that permitted the traffic.	

Field	Values	Additional information
Timeout	Idle timeout after which the session expires.	
In	Incoming flow (source and destination IP addresses, application protocol, and interface).	
Out	Reverse flow (source and destination IP addresses, application protocol, and interface).	

Monitoring flow session destination prefix information

To view information about each session that uses the specified destination prefix, select **Monitor** → **Security** → **Flow Session Statistics** in the J-Web interface, see Figure 10-32 on page 365. Then select **destination prefix** from the Session Filter list and click **Show**. Alternatively, enter the following CLI command:

```
show security flow session destination-prefix destination-prefix-number
```

Table 10-17 summarizes key output fields in the flow session destination prefix display.

Table 10-17 Summary of key flow destination prefix session information output fields

Field	Values	Additional information
Flow session statistics: session filter—destination prefix		
Session ID	Number that identifies the session. Use this ID to get more information about the session.	
Policy name	Policy that permitted the traffic.	
Timeout	Idle timeout after which the session expires.	
In	Incoming flow (source and destination IP addresses, application protocol, and interface).	
Out	Reverse flow (source and destination IP addresses, application protocol, and interface).	

Monitoring flow session interface information

To view information about each session that uses the specified incoming or outgoing interface, select **Monitor** → **Security** → **Flow Session Statistics** in the J-Web interface, see Figure 10-32 on page 365. Then select **interface** from the Session Filter list and click **Show**. Alternatively, enter the following CLI command:

```
show security flow session interface interface-name
```

Table 10-18 summarizes key output fields in the flow session interface display.

Table 10-18 Summary of key flow interface session information output fields

Field	Values	Additional information
Flow session statistics: session filter—interface		
Session ID	Number that identifies the session. Use this ID to get more information about the session.	
Policy name	Policy that permitted the traffic.	
Timeout	Idle timeout after which the session expires.	

Field	Values	Additional information
In	Incoming flow (source and destination IP addresses, application protocol, and interface).	
Out	Reverse flow (source and destination IP addresses, application protocol, and interface).	

Monitoring flow session protocol information

To view information about each session that uses the specified protocol, select **Monitor** → **Security** → **Flow Session Statistics** in the J-Web interface, see Figure 10-32 on page 365. Then select **protocol** from the Session Filter list and click **Show**. Alternatively, enter the following CLI command:

```
show security flow session protocol protocol-name
```

Table 10-19 1 summarizes key output fields in the flow session protocol display.

Table 10-19 Summary of key flow protocol session information output fields

Field	Values	Additional information
Flow session statistics: session filter—protocol		
Session ID	Number that identifies the session. Use this ID to get more information about the session.	
Policy name	Policy that permitted the traffic.	
Timeout	Idle timeout after which the session expires.	
In	Incoming flow (source and destination IP addresses, application protocol, and interface).	
Out	Reverse flow (source and destination IP addresses, application protocol, and interface).	

Monitoring flow session resource manager

To view information about sessions created by the resource manager, select **Monitor** → **Security** → **Flow Session Statistics** in the J-Web interface, see Figure 10-32 on page 365. Then select **resource manager** from the Session Filter list and click **Show**. Alternatively, enter the following CLI command:

```
show security flow session resource-manager
```

Table 10-20 summarizes key output fields in the flow session resource manager display.

Table 10-20 Summary of key flow resource manager session output fields

Field	Values	Additional information
Flow session statistics: session filter—resource manager		
Session ID	Number that identifies the session. Use this ID to get more information about the session.	
Policy name	Policy that permitted the traffic.	
Timeout	Idle timeout after which the session expires.	

Field	Values	Additional information
Resource information	Information about the session particular to the resource manager, including the name of the ALG, the group ID. and the resource ID.	
Flow session statistics: Session ID		
In	Incoming flow (source and destination IP addresses, application protocol, and interface).	
Out	Reverse flow (source and destination IP addresses, application protocol, and interface).	

Monitoring flow session identifier session

To view information about the session, select **Monitor** → **Security** → **Flow Session Statistics** in the J-Web interface, see Figure 10-32 on page 365. Then select **session identifier** from the Session Filter list and click **Show**. Alternatively, enter the following CLI command:

```
show security flow session session-identifier session-identifier
```

Table 10-21 summarizes key output fields in the flow session identifier session display.

Table 10-21 Summary of key flow session identifier output fields

Field	Values	Additional information
Flow Session Statistics: session filter—session identifier		
Session ID	Number that identifies the session. Use this ID to get more information about the session.	
Status	Session status	
Flag	Internal flag depicting the state of the session, used for debugging purposes.	
Virtual system	Virtual system to which the session belongs.	
Policy name	Name and ID of the policy that the first packet of the session matched.	
Maximum timeout	Maximum session timeout.	
Current timeout	Remaining time for the session unless traffic exists in the session.	
Start time	Time when the session was created, offset from the system start time.	
Duration	Length of time for which the session is active.	

Field	Values	Additional information
In	For the input flow: <ul style="list-style-type: none"> ▶ Source and destination addresses and protocol tuple for the input flow. ▶ Interface: Input flow interface. ▶ Session token: Internal token derived from the virtual routing instance. ▶ Flag: Internal debugging flags. ▶ Route: Internal next hop of the route to be used by the flow. ▶ Gateway: Next-hop gateway of the flow. ▶ Tunnel: If the flow is going into a tunnel, the tunnel ID. Otherwise, 0 (zero). ▶ Port Sequence, FIN sequence, FIN state, Cookie: Internal TCP state tracking information. 	
Out	For the reverse flow: <ul style="list-style-type: none"> ▶ Source and destination addresses and protocol tuple for the input flow. ▶ Interface: Input flow interface. ▶ Session token: Internal token derived from the virtual routing instance. ▶ Flag: Internal debugging flags. ▶ Route: Internal next hop of the route to be used by the flow. ▶ Gateway: Next-hop gateway of the flow. ▶ Tunnel: If the flow is going into a tunnel, the tunnel ID. Otherwise, 0 (zero). ▶ Port Sequence, FIN sequence, FIN state, Cookie: Internal TCP state tracking information. 	

Monitoring flow session source port information

To view information about each session that uses the specified source port, select **Monitor** → **Security** → **Flow Session Statistics** in the J-Web interface, see Figure 10-32 on page 365. Then select **source port** from the Session Filter list and click **Show**. Alternatively, enter the following CLI command:

```
show security flow session source-port source-port-number
```

Table 10-22 summarizes key output fields in the flow session source port display.

Table 10-22 Summary of key flow source port session output fields

Field	Values	Additional information
Flow session statistics: session filter—source port		
Session ID	Number that identifies the session. Use this ID to get more information about the session.	
Policy name	Policy that permitted the traffic.	
Timeout	Idle timeout after which the session expires.	
In	Incoming flow (source and destination IP addresses, application protocol, and interface).	

Field	Values	Additional information
Out	Reverse flow (source and destination IP addresses, application protocol, and interface).	

Monitoring flow session source prefix information

To view information about each session that uses the specified source prefix, select **Monitor** → **Security** → **Flow Session Statistics** in the J-Web interface, see Figure 10-32 on page 365. Then select **source prefix** from the Session Filter list and click **Show**. Alternatively, enter the following CLI command:

```
show security flow session source-prefix source-prefix-number
```

Table 10-23 summarizes key output fields in the flow session source prefix display.

Table 10-23 Summary of key flow source prefix session output fields

Field	Values	Additional information
Flow session statistics: session filter—source prefix		
Session ID	Number that identifies the session. Use this ID to get more information about the session.	
Policy name	Policy that permitted the traffic.	
Timeout	Idle timeout after which the session expires.	
In	Incoming flow (source and destination IP addresses, application protocol, and interface).	
Out	Reverse flow (source and destination IP addresses, application protocol, and interface).	

Monitoring flow session tunnel information

To view information about all tunnel session, select **Monitor** → **Security** → **Flow Session Statistics** in the J-Web interface. Then select **tunnel** from the Session Filter list, and click **Show**. Alternatively, enter the following CLI command:

```
show security flow session tunnel
```

Table 10-24 summarizes key output fields in the flow session tunnel display.

Table 10-24 Summary of key flow tunnel session output fields

Field	Values	Additional information
Flow session statistics: session filter—tunnel		
Session ID	Number that identifies the session. Use this ID to get more information about the session.	
Policy name	Policy that permitted the traffic.	
Timeout	Idle timeout after which the session expires.	
In	Incoming flow (source and destination IP addresses, application protocol, and interface).	
Out	Reverse flow (source and destination IP addresses, application protocol, and interface).	

10.2.9 Monitoring flow gate information

To view information about temporary openings known as pinholes or gates in the security firewall, select **Monitor** → **Security** → **Flow Gate** Information in the J-Web interface, or enter the following CLI command:

```
show security flow gate
```

Table 10-25 summarizes key output fields in the flow gate display.

Table 10-25 Summary of key flow tunnel session output fields

Field	Values	Additional information
Flow gate information		
Hole	Range of flows permitted by the pinhole.	
Translated	Tuples used to create the session if it matches the pinhole: <ul style="list-style-type: none">► Source address and port► Destination address and port	
Protocol	Application protocol, such as UDP or TCP.	
Application	Name of the application.	
Age	Idle timeout for the pinhole.	
Flags	Internal debug flags for pinhole.	
Zone	Incoming zone.	
Reference count	Number of resource manager references to the pinhole.	
Resource	Resource manager information about the pinhole.	

10.2.10 Monitoring firewall authentication

The J-Web interface provides information about user authentications and history of authentications.

Monitoring firewall authentication table

The firewall authentication user information is divided into multiple parts. To view information about authentication table, select **Monitor** → **Security** → **Firewall Authentication** → **Authentication Table**, in the J-Web interface, as shown in Figure 10-33 on page 374. To view detailed information about the user with a particular identifier, select the ID on the Authentication Table page. To view detailed information about the user at a particular source IP address, select the Source IP on the Authentication Table page.

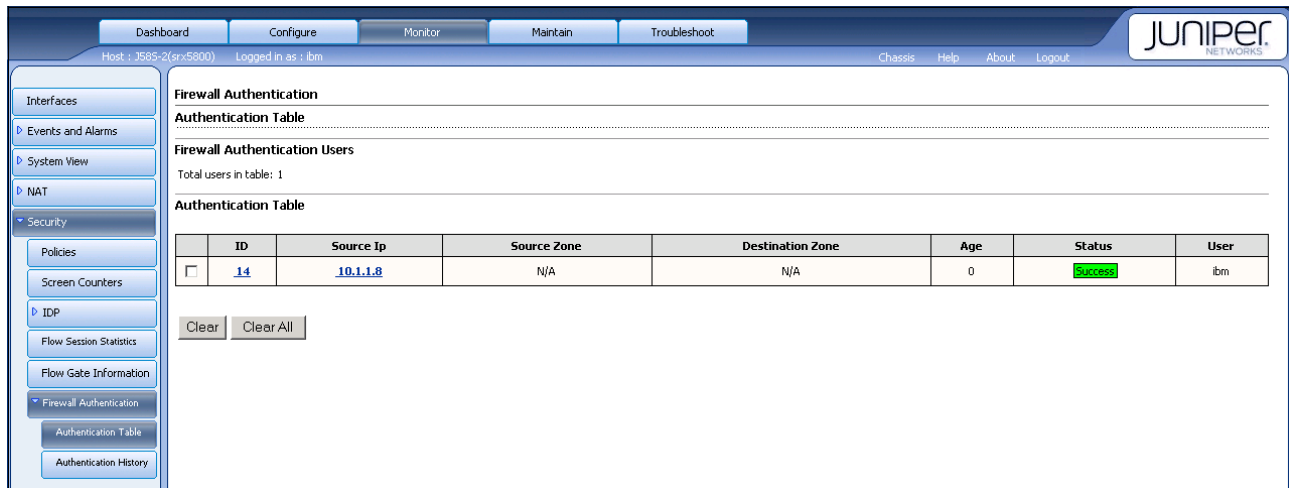


Figure 10-33 J-Web Firewall Authentication

Alternatively, enter the following CLI commands:

```
show security firewall-authentication users
show security firewall-authentication users address ip-address
show security firewall-authentication users identifier identifier
```

Table 10-26 summarizes key output fields in firewall authentication table display.

Table 10-26 Summary of Key Flow Tunnel Session Output Fields

Field	Values	Additional information
Firewall authentication users		
Total users in table	Number of users in the authentication table.	
Authentication table		
ID	Authentication identification number.	
Source Ip	IP address of the authentication source.	
Age	Idle timeout for the user.	
Status	Status of authentication (success or failure).	
user	Name of the user.	
Detailed report per ID selected: <i>ID</i>		
Source Zone	Name of the source zone.	
Destination Zone	Name of the destination zone.	
profile	Name of the profile.	Users information.
Authentication method	Path chosen for authentication.	
Policy Id	Policy Identifier.	

Field	Values	Additional information
Interface name	Name of the interface.	
Bytes sent by this user	Number of packets in bytes sent by this user.	
Bytes received by this user	Number of packets in bytes received by this user.	
Client-groups	Name of the client group.	
Detailed report per Source Ip selected		
Entries from Source IP	IP address of the authentication source.	
Source Zone	Name of the source zone.	
Destination Zone	Name of the destination zone.	
profile	Name of the profile.	
Age	Idle timeout for the user.	
Status	Status of authentication (success or failure).	
user	Name of the user.	
Authentication method	Path chosen for authentication.	
Policy Id	Policy Identifier.	
Interface name	Name of the interface.	
Bytes sent by this user	Number of packets in bytes sent by this user.	
Bytes received by this user	Number of packets in bytes received by this user.	
Client-groups	Name of the client group.	

Monitoring firewall authentication history

The firewall authentication history information is divided into multiple parts. To view information about the authentication history, select **Monitor** → **Security** → **Firewall Authentication** → **Authentication History** in the J-Web interface, as shown in Figure 10-34 on page 376. To view the detailed history of the authentication with this identifier, select the ID on the Firewall Authentication History page. To view a detailed authentication history of this source IP address, select the Source IP on the Firewall Authentication History page.

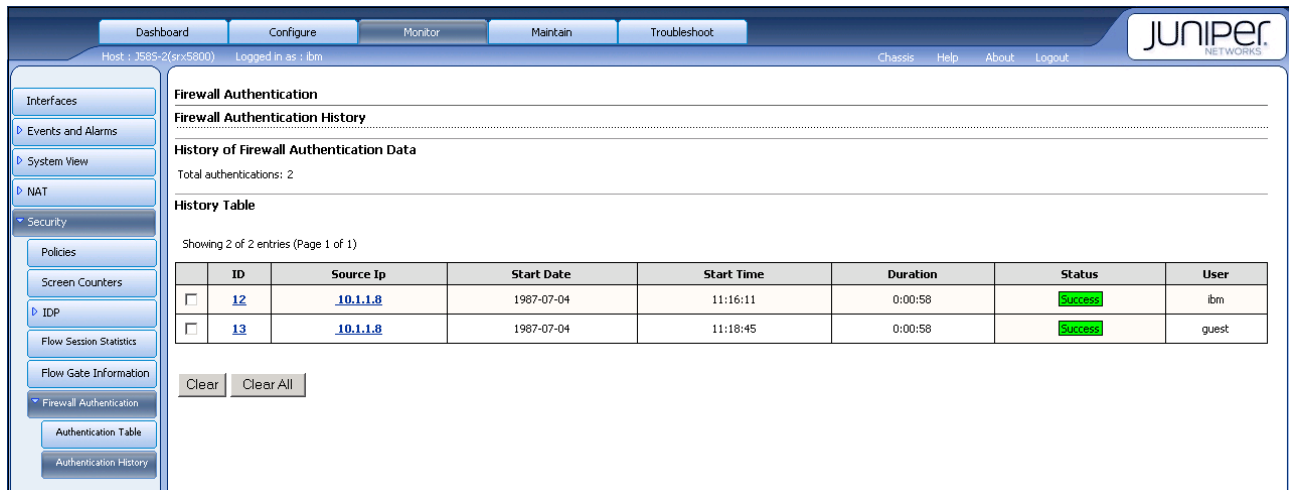


Figure 10-34 J-Web Firewall Authentication History

Alternatively, enter the following CLI show commands:

show security firewall-authentication history

show security firewall-authentication history address *ip-address*

show security firewall-authentication history identifier *identifier*

Table 10-27 summarizes key output fields in firewall authentication history display.

Table 10-27 Summary of Key Firewall Authentication History output fields

Field	Values	Additional information
History of Firewall Authentication Data		
Total authentications	Number of authentication	
History Table		
ID	Identification number.	
Source Ip	IP address of the authentication source.	
Start Date	Authentication date.	
Start Time	Authentication time.	
Duration	Authentication duration.	
Status	Status of authentication (success or failure).	
User	Name of the user.	
Detail history of selected Id: <i>ID</i>		
Authentication method	Path chosen for authentication.	
Policy Id	Security policy identifier.	
Source zone	Name of the source zone.	
Destination Zone	Name of the destination zone.	

Field	Values	Additional information
Interface name	Name of the interface.	
Bytes sent by this user	Number of packets in bytes sent by this user.	
Bytes received by this user	Number of packets in bytes received by this user.	
Client-groups	Name of the client group.	
Detail history of selected Source Ip: <i>Source Ip</i>		
User	Name of the user.	
Start Date	Authentication date.	
Start Time	Authentication time.	
Duration	Authentication duration.	
Status	Status of authentication (success or failure).	
Profile	Name of the profile.	
Authentication method	Path chosen for authentication.	
Policy Id	Security policy identifier.	
Source zone	Name of the source zone.	
Destination Zone	Name of the destination zone.	
Interface name	Name of the interface.	
Bytes sent by this user	Number of packets in bytes sent by this user.	
Bytes received by this user	Number of packets in bytes received by this user.	
Client-groups	Name of the client group.	

10.3 Monitoring events and managing system log files

Junos software supports configuring and monitoring of system log messages (also called syslog messages). You can configure files to log system messages and also assign attributes, such as severity levels, to messages. The View Events page on the J-Web interface enables you to filter and view system log messages. For more information about system log messages, see the *JUNOS Software System Log Messages Reference GA32-0675-00*.

10.3.1 System log message terms

Before configuring and monitoring system log messages, become familiar with the terms defined in Table 10-28 on page 378.

Table 10-28 System Log Message Terms

Term	Definition
event	Condition that occurs on a device at a particular time. An event can include routine, failure, error, emergency or critical conditions.
event ID	System log message code that uniquely identifies a system log message. The code begins with a prefix indicating the software process or library that generates the event.
facility	Group of messages that either are generated by the same software process (such as accounting statistics) or concern a similar condition or activity (such as authentication attempts). For a list of system logging facilities, see Table 10-29 on page 381.
Priority	Combination of the facility and severity level of a system log message. By default, priority information is not included in system log messages, but you can configure Junos Software to include it. For more information, see the <i>JUNOS Software System Log Messages Reference GA32-0675-00</i> .

Term	Definition
process	<p>Software program, also known as a daemon, that controls device functionality. The following are the primary Junos Software processes:</p> <ul style="list-style-type: none"> ▶ Routing protocol process (rpd): Defines how routing protocols such as RIP, OSPF, and BGP operate on the device. It starts the configured routing protocols, handles all routing messages, maintains routing tables and implements the routing policy. ▶ Interface process (dcd): Allows you to configure and control the physical and logical interfaces present in a device. It also enables Junos Software to track the status and condition of the device's interfaces. ▶ Chassis process (chassisd): Controls the physical properties of a device chassis, including conditions that trigger alarms. ▶ SNMP—Simple Network Management Protocol, which helps administrators monitor the state of a device. ▶ Management process (mgd): Controls processes that start and monitor all the other software processes. The management process starts the command-line interface (CLI), which is the primary tool used to control and monitor Junos Software. It also starts all the software processes and the CLI when the device starts up. If a software process terminates, the management process attempts to restart it. ▶ Forwarding process (flowd): Forwards packets through the device. The flow-based forwarding process applies filters and policers associated with the ingress interface to packets entering the device. It establishes the state of the packet's session and manages the packet as it transits the security flow and its applicable features. It applies output filtering and traffic shaping to the flow before transmitting the packet out the egress interface. ▶ Network security process (nsd): Interprets, executes, and manages the configuration of extended interface attributes, policies, zones, address books, firewall screens, Network Address Translation (NAT), and other network security treatments. ▶ Internet Key Exchange process (iked): Implements tunnel management for IPSec VPNs, provides authentication of endpoint entities, and generates keys for packet authentication and encryption. ▶ Firewall authentication process (fwauthd): Implements and manages user authentication configuration, and authenticates users who access the firewall. ▶ Dynamic Host Configuration Protocol process (dhcpcd): Implements the DHCP client, allowing the device to obtain IP addresses from the network DHCP server, set other configuration parameters, manage TCP/IP settings propagation, and display client-related information.
process ID	Identifier uniquely identifying a process. The process ID is displayed in a system log message along with the name of the process that generates the event.
severity level	Measure of how seriously a triggering event affects device functions. For a list of severity levels that you can specify, see Table 10-30 on page 381.

10.3.2 System log messages overview

Junos Software generates system log messages to record events that occur on the device, including the following:

- ▶ Routine operations, such as creation of an Open Shortest Path First (OSPF) protocol adjacency or a user login into the configuration database
- ▶ Failure and error conditions, such as failure to access a configuration file or unexpected closure of a connection to a child or peer process
- ▶ Emergency or critical conditions, such as device power-off due to excessive temperature

The Junos system logging utility is similar to the UNIX *syslogd* utility. Each system log message identifies the software process that generated the message and briefly describes the operation or error that occurred. The default log directory for all system log messages is `/cf/var/log`.

Reboot requests are recorded to the system log files, which you can view with the **show log** command. Also, you can view the names of any processes running on your system with the **show system processes** command.

System log message destinations

You can send system logging information to one or more destinations. The destinations can be one or more files, one or more remote hosts, the terminals of one or more users if they are logged in, and the system console.

- ▶ To direct messages to a named file in a local file system, see “Sending system log messages to a file” on page 383.
- ▶ To direct messages to the terminal session of one or more specific users (or all users) when they are logged into the device, see “Sending system log messages to a user terminal” on page 384.
- ▶ To send a security log stream to a remote server, see “Setting the system to stream security logs through revenue ports” on page 383
- ▶ To direct messages to the device console, see the *JUNOS Software System Log Messages Reference GA32-0675-00*.
- ▶ To direct messages to a remote machine that is running the UNIX *syslogd* utility, see the *JUNOS Software System Log Messages Reference, GA32-0675*.

Redundant system log server

Security system logging traffic intended for remote servers is sent through the network interface ports, which support two simultaneous system log destinations. Each system logging destination must be configured separately, see “Setting the system to stream security logs through revenue ports” on page 383. When two system log destination addresses are configured, identical logs are sent to both destinations. While two destinations can be configured on any device that supports the feature, adding a second destination is primarily useful as a redundant backup for standalone, active/active, and active/backup configured chassis cluster deployments.

In chassis cluster deployments, since each node has its own logging mechanism, it is required to collect system logs on both nodes separately. It is also recommended that you use external system log servers since, on high end IBM j-type Ethernet Appliances, the system logs are sent at very high rate from the dataplane directly to the external syslog server.

System log facilities and severity levels

When specifying the destination for system log messages, you can specify the class (facility) of messages to log and the minimum severity level (level) of the message for each location.

Each system log message belongs to a facility, which is a group of messages that are either generated by the same software process or concern a similar condition or activity.

Table 10-29 lists the system logging facilities, and Table 10-30 lists the system logging severity levels. For more information about system log messages, see the *JUNOS Software System Log Messages Reference GA32-0675-00*.

Table 10-29 System logging facilities

Facility	Description
any	Any facility
authorization	Any authorization attempt
change-log	Any change to the configuration
cron	Cron scheduling process
daemon	Various system processes
interactive-commands	Commands executed in the CLI
kernel	Messages generated by the Junos kernel
user	Messages from random user processes

Table 10-30 System logging severity levels

Severity level (from highest to lowest severity)	Description
emergency	System panic or other conditions that cause the routing platform to stop functioning.
alert	Conditions that must be corrected immediately, such as a corrupted system database.
critical	Critical conditions, such as hard drive errors.
error	Standard error conditions that generally have less serious consequences than errors in the emergency, alert, and critical levels.
warning	Conditions that warrant monitoring.
notice	Conditions that are not error conditions but are of interest or might warrant special handling.
info	Informational messages. This is the default.
debug	Software debugging messages.

Control plane and data plane logs

Junos Software generates separate log messages to record events that occur on the system's control and data planes:

- The control plane logs include events that occur on the routing platform. The system sends control plane events to the eventd process on the Routing Engine, which then

handles the events by using Junos policies and/or by generating system log messages. You can choose to send control plane logs to a file, user terminal, routing platform console, or remote machine. To generate control plane logs, use the **syslog** statement at the **[system]** hierarchy level

- The data plane logs primarily include security events that the system has handled directly inside the data plane. These system logs are also referred to as security logs. How the system handles data plane events depends on the device:

For the IBM j-type Ethernet Appliances, by default, the system streams already-processed data plane events directly to external log servers, bypassing the Routing Engine. If an event requires processing, the system sends the event to the *eventd* process on the Routing Engine.

For information about changing these settings, see “Setting the system to send all log messages through event” on page 382 and “Setting the system to stream security logs through revenue ports” on page 383.

10.3.3 Configuring system log messages with a configuration editor

In this section, we discuss how to configure system log messages with a configuration editor.

Setting the system to send all log messages through event

To have security logs handled by the *eventd* process and sent with system logs to a remote server, enter the following command:

```
{primary:node0}
ibm@J58S> set security log mode event
```

Then configure the server that will receive the system log messages:

```
{primary:node0}
ibm@J58S> set system syslog host hostname
```

In the command, *hostname* is the fully qualified host name or IP address of the server that will receive the logs.

This type of logging configuration is the one that has been used most commonly for Junos. In this configuration, control plane logs and data plane, or security, logs are forwarded from the data plane to the Routing Engine control plane *rtlogd* process. The *rtlogd* process then either forwards syslog/sd-syslog-formatted logs to the *eventd* process or the WELF-formatted logs to the external/remote WELF log collector.

Note: If you want to send duplicate logs to a second remote server, repeat the command with a new fully qualified hostname or IP address of a second server. If your deployment is an active/active chassis cluster, you can also configure security logging on the active node to be sent to separate remote servers in order to achieve logging redundancy

If you need to rename or redirect one of the logging configurations, you will need to delete and recreate it. To delete a configuration:

```
{primary:node0}
ibm@J58S-1> delete security log mode event hostname
```

Setting the system to stream security logs through revenue ports

Note: WELF logs must be streamed through a revenue port because the *eventd* process does not recognize the WELF format.

You can increase the number of data plane, or security, logs that are sent by modifying the manner in which they are sent.

When the logging mode is set to stream, security logs generated in the data plane are streamed out a revenue traffic port directly to a remote server. Other system logs are still handled as described in “Setting the system to send all log messages through event” on page 382.

To use the stream mode, enter the following commands:

```
{primary:node0}
ibm@J58S-1> set security log mode source-address
{primary:node0}
ibm@J58S-1> set security log mode stream
{primary:node0}
ibm@J58S-1> set security log stream streamname format [syslog | sd-syslog |
welf] category [all | content-security] host ipaddr
```

Where *source-address* is the IP address of the source machine; **syslog**, **sd-syslog** (structured system logging messages), and **welf** are the logging formats; **all** and **content-security** are the categories of logging; and *ipaddr* is the IP address of the server to which the logs will be streamed.

For the WELF format, the category must be set to content-security. For example:

```
{primary:node0}
ibm@J58S-1> set security log stream securitylog1 format welf category
content-security host 10.121.23.5
```

Note: If you want to send duplicate logs to a second remote server, repeat the command with a new *ipaddr*.

If your deployment is an active/active chassis cluster, you can also configure security logging on the active node to be sent to separate remote servers in order to achieve logging redundancy.

Sending system log messages to a file

You can direct system log messages to a file on the CompactFlash card. The default directory for log files is */var/log*. To specify a different directory on the CompactFlash card, include the complete pathname. For the list of logging facilities and severity levels, see Table 10-29 on page 381 and Table 10-30 on page 381.

For information about archiving log files, see “Archiving system logs” on page 386. The procedure provided in this section sends all security-related information to the sample file named *security*.

To send messages to a file:

1. Navigate to the top of the configuration hierarchy in either the J-Web or CLI configuration editor.
2. Perform the configuration tasks described in Table 10-31.
3. If you are finished configuring the network, commit the configuration.

Table 10-31 Sending System Log Messages to a File

Task	J-Web Configuration Editor	CLI Configuration Editor
Navigate to the Syslog level in the configuration hierarchy.	<ol style="list-style-type: none"> 1. In the J-Web interface, select CLI Tools → Point and Click CLI. 2. Next to System, click Configure or Edit. 3. Next to Syslog, click Configure or Edit. See Figure 10-35 	<p>From the [edit] hierarchy level, enter</p> <pre>{primary:node0}[edit] ibm@J58S-1> edit system syslog</pre>
Create a file named security , and send log messages of the authorization class at the severity level info to the file.	<ol style="list-style-type: none"> 1. Next to File, click Add new entry. 2. In the File name box, type security. 3. Next to Contents, click Add new entry. 4. In the Facility list, select authorization. 5. In the Level list, select info. 6. Click OK. 	<p>Set the filename and the facility and severity level:</p> <pre>{primary:node0}[edit] ibm@J58S-1> set file security authorization info</pre>

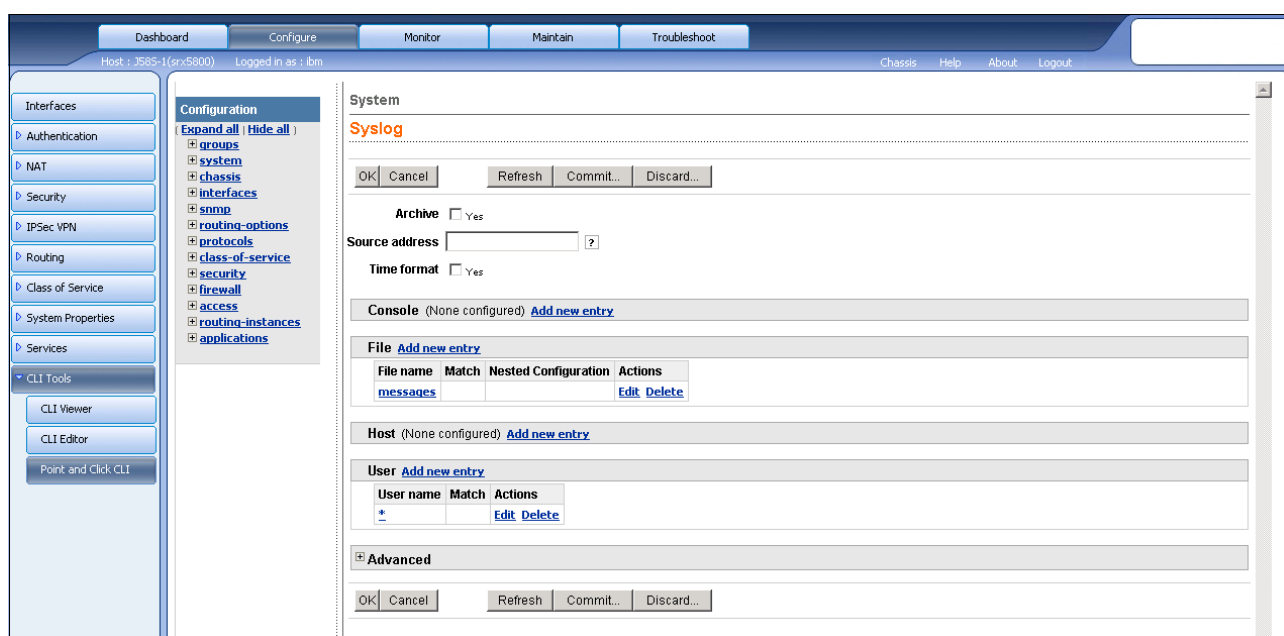


Figure 10-35 J-Web Syslog configuration

Sending system log messages to a user terminal

To direct system log messages to the terminal session of one or more specific users, or all users, when they are logged into the local Routing Engine, specify one or more Junos

usernames. Separate multiple values with spaces, or use the asterisk (*) to indicate all users who are logged into the local Routing Engine. For the list of logging facilities and severity levels, see Table 10-29 on page 381 and Table 10-30 on page 381.

The procedure provided in this section sends any critical messages to the terminal of the sample user frank, if he is logged in.

To send messages to a user terminal:

1. Navigate to the top of the configuration hierarchy in either the J-Web or CLI configuration editor.
2. Perform the configuration tasks described in Table 10-32.
3. If you are finished configuring the network, commit the configuration.

Table 10-32 Sending system log messages to a user

Task	J-Web Configuration Editor	CLI Configuration Editor
Navigate to the Syslog level in the configuration hierarchy.	<ol style="list-style-type: none"> 1. In the J-Web interface, select CLI Tools → Point and Click CLI. 2. Next to System, click Configure or Edit. 3. Next to Syslog, click Configure or Edit. See Figure 10-35 on page 384 	<p>From the [edit] hierarchy level, enter</p> <pre>{primary:node0}[edit] ibm@J58S-1> edit system syslog</pre>
Send all critical messages to the user frank.	<ol style="list-style-type: none"> 1. Next to User, click Add new entry. 2. In the User name box, type frank. 3. Next to Contents, click Add new entry. 4. In the Facility list, select any. 5. In the Level list, select critical. 6. Click OK. See Figure 10-36 on page 386 	<p>Set the User name and the facility and severity level:</p> <pre>{primary:node0}[edit] ibm@J58S-1> set user frank any critical</pre>

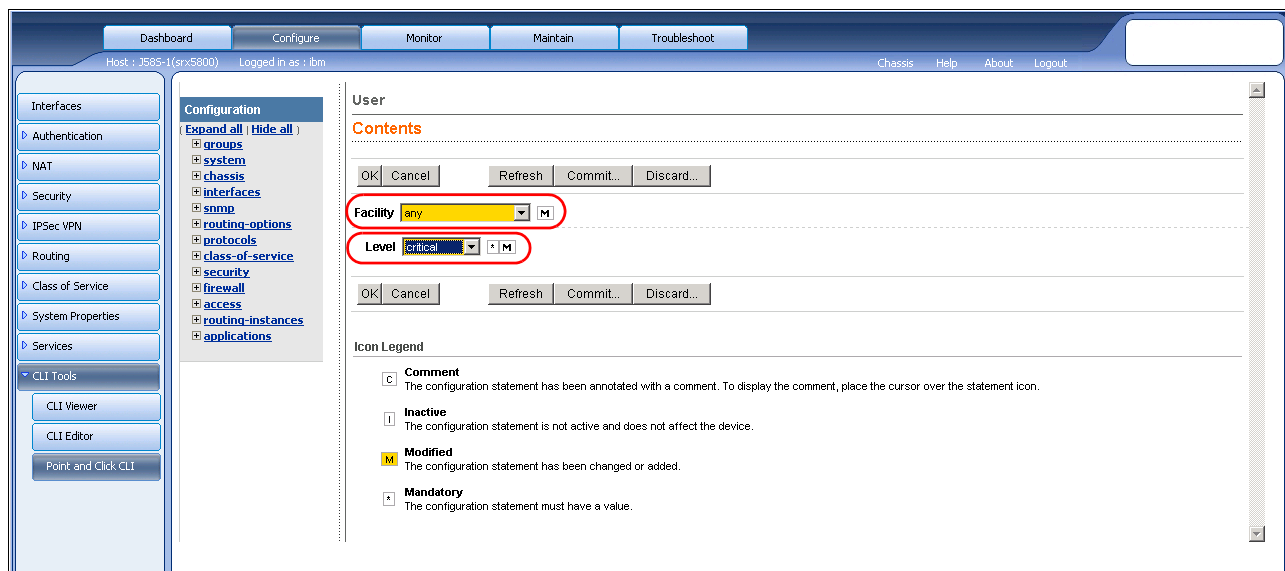


Figure 10-36 J-Web system log to user

Archiving system logs

By default, the Junos logging utility stops writing messages to a log file when the file reaches 128 KB in size. It closes the file and adds a numerical suffix, then opens and directs messages to a new file with the original name. By default, the logging utility creates up to 10 files before it begins overwriting the contents of the oldest file. The logging utility by default also limits the users who can read log files to the root user and users who have the Junos maintenance permission.

To enable all users to read log files, include the **world-readable** statement at the **[edit system syslog archive]** hierarchy level. To restore the default permissions, include the **no-world-readable** statement. You can include the **archive** statement at the **[edit system syslog file filename]** hierarchy level to configure the number of files, file size, and permissions for the specified log file. For configuration details, see the information about archiving log files in the *JUNOS Software System Basics and Services Command Reference GA32-0671-02*.

Disabling system logs

To disable logging of the messages from a facility, use the **facility none** configuration statement. This statement is useful when, for example, you want to log messages of the same severity level from all but a few facilities. Instead of including a configuration statement for each facility you want to log, you can configure the **any** level statement and then a **facility none** statement for each facility you do not want to log. For configuration details, see the information about disabling logging in the *JUNOS Software System Basics and Services Command Reference GA32-0671-02*.

10.3.4 Monitoring system log messages with the J-Web event viewer

To monitor errors and events that occur on the device, select **Monitor** → **Events and Alarms** → **View Events** in the J-Web user interface.

The J-Web View Events page, see Figure 10-37 on page 388, displays the following information about each event:

- Process: System process that generated the error or event.

- ▶ **Severity:** A severity level indicates how seriously the triggering event affects routing platform functions. Only messages from the facility that are rated at that level or higher are logged. Possible severities and their corresponding color code are:
 - **Debug/Info/Notice (Green):** Indicates conditions that are not errors but are of interest or might warrant special handling.
 - **Warning (Yellow):** Indicates conditions that warrant monitoring.
 - **Error (Blue):** Indicates standard error conditions that generally have less serious consequences than errors in the emergency, alert, and critical levels.
 - **Critical (Pink):** Indicates critical conditions, such as hard drive errors.
 - **Alert (Orange):** Indicates conditions that require immediate correction, such as a corrupted system database.
 - **Emergency (Red):** Indicates system panic or other conditions that cause the routing platform to stop functioning.
- ▶ **Event ID:** Unique ID of the error or event. The prefix on each code identifies the generating software process. The rest of the code indicates the specific event or error.
- ▶ **Event Description:** Displays a more detailed explanation of the message.
- ▶ **Time:** Time that the error or event occurred.

To control which errors and events are displayed in the list, use the following options:

- ▶ **System Log File:** Specify the name of the system log file that records the errors and events.
- ▶ **Process:** Specify the system processes that generate the events you want to display. For an overview of some of the primary system processes, see Table 10-28 on page 378. To view all the processes running on your system, enter the `show system processes CLI` command.
- ▶ **Date From:** Specify the beginning of the date range that you want to monitor. Set the date using the calendar pick tool.
- ▶ **To:** Specify the end of the date range that you want to monitor. Set the date using the calendar pick tool.
- ▶ **Event ID:** Specify the specific ID of the error or event that you want to monitor. For a complete list of system error and event IDs, see the *JUNOS Software System Log Messages Reference GA32-0675-00*.
- ▶ **Description:** Enter a description for the errors or events.
- ▶ **Search:** Fetches the errors and events specified in the search criteria.
- ▶ **Reset:** Clears the cache of errors and events that were previously selected.
- ▶ **Generate Report:** Creates an HTML report based on the specified parameters.

Events Filter

System Log File: messages

Process:

☐ Include archived files

Date From: 2010-05-25,13:34

To: 2010-05-25,14:34

Event ID:

Description:

Search

Reset

Events Detail

Generate Report

Process	Severity	Event ID	Event Description	Time
422			sc_process_ioctl failed	2010-05-25 13:50:30 UTC
handler			failed when processing message - type 1032 subtype 33 [generic failure]	2010-05-25 13:50:30 UTC
422			sc_process_ioctl failed	2010-05-25 13:50:30 UTC
kernel		GENCFG	op 2 (Gencfg Blob) failed; err 7 (Doesn't Exist)	2010-05-25 13:49:58 UTC
kernel		GENCFG	op 6 (Voyager IPSEC SA) failed; err 7 (Doesn't Exist)	2010-05-25 13:49:58 UTC
kernel		GENCFG	op 6 (Voyager IPSEC SA) failed; err 7 (Doesn't Exist)	2010-05-25 13:49:58 UTC
mgd	notice	UI_LOAD_EVENT	User 'Yibm' is performing a 'load patch'	2010-05-25 13:49:58 UTC
mgd	notice	UI_LOAD_EVENT	User 'Yibm' is performing a 'load patch'	2010-05-25 13:49:57 UTC
mgd	notice	UI_COMMIT	User 'Yibm' requested 'commit' operation (comment: none)	2010-05-25 13:49:57 UTC
stub			ukern_send_sess_scan_cmd[106]	2010-05-25 13:49:48 UTC
stub			ukern_send_sess_scan_cmd[106]	2010-05-25 13:49:48 UTC
kernel		GENCFG	op 2 (Gencfg Blob) failed; err 7 (Doesn't Exist)	2010-05-25 13:49:48 UTC
stub			ukern_send_sess_scan_cmd[106]	2010-05-25 13:49:48 UTC
mgd	notice	UI_LOAD_EVENT	User 'Yibm' is performing a 'load patch'	2010-05-25 13:49:48 UTC
mgd	notice	UI_LOAD_EVENT	User 'Yibm' is performing a 'load patch'	2010-05-25 13:49:47 UTC
mgd	notice	UI_COMMIT	User 'Yibm' requested 'commit' operation (comment: none)	2010-05-25 13:49:47 UTC
kernel		GENCFG	op 2 (Gencfg Blob) failed; err 7 (Doesn't Exist)	2010-05-25 13:48:49 UTC
kernel		RT_PFE	NH details: idx 559 type 2 ifl 94	2010-05-25 13:48:49 UTC
kernel		RT_PFE	NH IPC op 1 (ADD NEXTHOP) failed; err 1 (Unknown)	2010-05-25 13:48:49 UTC

Page 3 of 4

Complete (92 events loaded.)

Unknown

Debug

Info

Notice

Warning

Error

Critical

Alert

Emergency

Displaying 51 - 75 of 92

Figure 10-37 J-Web Event Viewer

10.4 Configuring and monitoring alarms

Alarms alert you to conditions on a network interface, on the device chassis, or in the system software that might prevent the device from operating normally. You can set the conditions that trigger alarms on an interface. Chassis and system alarm conditions are preset.

An active alarm lights the ALARM LED on the front panel of the device. You can monitor active alarms from the J-Web interface or the CLI. For more information about alarms, see the *JUNOS Software System Basics and Services Command Reference GA32-0671-02*.

10.4.1 Alarm terms

Before configuring and monitoring alarms, become familiar with the terms defined in Table 10-33.

Table 10-33 Alarm terms

Term	Definition
Alarm	Signal alerting you to conditions that might prevent normal operation. The alarm signal is the yellow ALARM LED lit on the front of the chassis.

Term	Definition
Alarm condition	Failure event that triggers an alarm.
Alarm severity	Seriousness of the alarm. The level of severity can be either major (red) or minor (yellow).
Chassis alarm	Predefined alarm triggered by a physical condition on the device such as a power supply failure, excessive component temperature, or media failure.
Interface alarm	Alarm triggered by the state of a physical link on a fixed or installed Physical Interface Module (PIM), such as a link failure or a missing signal. Interface alarms are triggered by conditions on a T1 (DS1), Fast Ethernet, serial, or T3 (DS3) physical interface or by conditions on the sp-0/0/0 adaptive services interface for stateful firewall filter, Network Address Translation (NAT), Intrusion Detection and Prevention (IDP), or IP Security (IPsec) services. To enable an interface alarm, you must explicitly set an alarm condition.
System alarm	Predefined alarm triggered by a missing rescue configuration or failure to install a license for a licensed software feature.

10.4.2 Alarm overview

Alarms warn you about conditions that can prevent the device from operating normally.

When an alarm condition triggers an alarm, the device lights the yellow (amber) **ALARM** LED on the front panel. When the condition is corrected, the light turns off.

Alarm types

The device supports three types of alarms:

- ▶ Interface alarms indicate a problem in the state of the physical links on fixed or installed PIMs. To enable interface alarms, you must configure them.
- ▶ Chassis alarms indicate a failure on the device or one of its components. Chassis alarms are preset and cannot be modified.
- ▶ System alarms indicate a missing rescue configuration or software license, where valid. System alarms are preset and cannot be modified, although you can configure them to appear automatically in the J-Web or CLI display.

Alarm severity

Alarms have two severity levels:

- ▶ Major (red): Indicates a critical situation on the device that has resulted from one of the following conditions. A red alarm condition requires immediate action:
 - One or more hardware components have failed.
 - One or more hardware components have exceeded temperature thresholds.
 - An alarm condition configured on an interface has triggered a critical warning.
- ▶ Minor (yellow): Indicates a noncritical condition on the device that, if left unchecked, might cause an interruption in service or degradation in performance. A yellow alarm condition requires monitoring or maintenance.

A missing rescue configuration or software license generates a yellow system alarm.

Alarm conditions

To enable alarms on a device interface, you must select an alarm condition and an alarm severity. In contrast, alarm conditions and severity are preconfigured for chassis alarms and system alarms.

Note: For information about chassis alarms for your device, see Chapter 11, “Maintenance and analysis” on page 401.

Interface alarm conditions

Table 10-34 lists the interface conditions, sorted by interface type, that you can configure for an alarm. Each alarm condition can be configured to trigger either a major (red) alarm or minor a (yellow) alarm. The corresponding configuration option is included.

For the services stateful firewall filters (NAT, IDP, and IPsec), which operate on an internal adaptive services module within a device, you can configure alarm conditions on the integrated services and services interfaces.

Table 10-34 Interface alarm conditions

Interface	Alarm condition	Description	Configuration option
DS1 (T1)	Alarm indication signal (AIS)	The normal T1 traffic signal contained a defect condition and has been replaced by the AIS. A transmission interruption occurred at the remote endpoint or upstream of the remote endpoint. This all-ones signal is transmitted to prevent consequential downstream failures or alarms.	ais
	Yellow alarm	The remote endpoint is in yellow alarm failure. This condition is also known as a far-end alarm failure.	ylw
Ethernet	Link is down	The physical link is unavailable.	link-down
Integrated Services	Hardware or software failure	On the adaptive services module, either the hardware associated with the module or the software that drives the module has failed.	failure

Interface	Alarm condition	Description	Configuration option
Serial	Clear-to-send (CTS) signal absent	The remote endpoint of the serial link is not transmitting a CTS signal. The CTS signal must be present before data can be transmitted across a serial link.	cts-absent
	Data carrier detect (DCD) signal absent	The remote endpoint of the serial link is not transmitting a DCD signal. Because the DCD signal transmits the state of the device, no signal probably indicates that the remote endpoint of the serial link is unavailable.	dcd-absent
	Data set ready (DSR) signal absent	The remote endpoint of the serial link is not transmitting a DSR signal. The DSR signal indicates that the remote endpoint is ready to receive and transmit data across the serial link.	dsr-absent
	Loss of receive clock	The clock signal from the remote endpoint is not present. Serial connections require clock signals to be transmitted from one endpoint and received by the other endpoint of the link.	loss-of-rx-clock
	Loss of transmit clock	The local clock signal is not present. Serial connections require clock signals to be transmitted from one endpoint and received by the other endpoint of the link.	loss-of-tx-clock

Interface	Alarm condition	Description	Configuration option
Services	Services module hardware down	A hardware problem has occurred on the device's services module. This error typically means that one or more of the CPOS on the module has failed.	hw-down
	Services link down	The link between the device and its services module is unavailable	linkdown
	Services module held in reset	The device's services module is stuck in reset mode. If the services module fails to start up five or more times in a row, the services module is held in reset mode. Startup fails when the amount of time from CPU release to CPU halt is less than 300 seconds.	pic-hold-reset
	Services module reset	The device's services module is resetting. The module resets after it crashes or is reset from the CLI, or when it takes longer than 60 seconds to start up.	pic-reset
	Services software module down	A software problem has occurred on the device's services module.	sw-down
E3	Alarm indication signal (AIS)	The normal E3 traffic signal contained a defect condition and has been replaced by the AIS. A transmission interruption occurred at the remote endpoint or upstream of the remote endpoint. This all-ones signal is transmitted to prevent consequential downstream failures or alarms.	ais
	Loss of signal (LOS)	No remote E3 signal is being received at the E3 interface.	los
	Out of frame (OOF)	An OOF condition has existed for 10 seconds. This alarm applies only to E3 interfaces configured in frame mode. The OOF failure is cleared when no OOF or LOS defects have occurred for 20 seconds.	oof
	Remote defect indication	An AIS, LOS, or OOF condition exists. This alarm applies only to E3 interfaces configured in frame mode.	rdi

Interface	Alarm condition	Description	Configuration option
T3 (DS3)	Alarm indication signal	The normal T3 traffic signal contained a defect condition and has been replaced by the AIS. A transmission interruption occurred at the remote endpoint or upstream of the remote endpoint. This all-ones signal is transmitted to prevent consequential downstream failures or alarms.	ais
	Excessive number of zeros	The bit stream received from the upstream host has more consecutive zeros than are allowed in a T3 frame.	exz
	Far-end receive failure (FERF)	The remote endpoint of the connection has failed. A FERG differs from a yellow alarm, because the failure can be any failure, not just an OOF or LOS failure.	ferf
	Idle alarm	The Idle signal is being received from the remote endpoint.	idle
	Line code violation	Either the line encoding along the T3 link is corrupted or a mismatch between the encoding at the local and remote endpoints of a T3 connection occurred.	lcv
	Loss of frame (LOF)	An OOF or loss-of-signal LOS condition has existed for 10 seconds. The LOF failure is cleared when no OOF or LOS defects have occurred for 20 seconds. A LOF failure is also called a red failure.	lof
	Loss of signal	No remote T3 signal is being received at the T3 interface	los
	Phase-locked loop out of lock	The clocking signals for the local and remote endpoints no longer operate in lock-step.	pll
	Yellow alarm	The remote endpoint is in yellow alarm failure. This condition is also known as a far-end alarm failure.	ylw

System alarm conditions and corrective actions

Table 10-35 on page 394 lists the two preset system alarms, the condition that triggers each alarm, and the action you take to correct the condition.

Table 10-35 System alarm conditions and corrective actions

Alarm type	Alarm condition	Corrective action
Configuration	The rescue configuration is not set.	Set the rescue configuration. For instructions, see the <i>JUNOS Software CLI User Guide</i> , GA32-0697-00.
License	<p>You configured at least one software feature that requires a feature license, but no valid license for the feature is currently installed.</p> <p>NOTE: This alarm indicates that you are in violation of the software license agreement. You must install a valid license key to be in compliance with all agreements.</p>	Install a valid license key. For instructions, see the Chapter 11, “Maintenance and analysis” on page 401.

10.4.3 Configuring alarms with a configuration editor

To configure interface alarms on a device, you must select the network interface on which to apply an alarm and the condition you want to trigger the alarm. For a list of conditions, see “Interface alarm conditions” on page 390.

To configure interface alarms:

1. Navigate to the top of the configuration hierarchy in either the J-Web or CLI configuration editor.
2. Perform the configuration tasks described in Table 10-36.
3. If you are finished configuring the network, commit the configuration.
4. To verify the alarms configuration, see “Verifying the alarms configuration” on page 398.
5. To check the status of active alarms, see “Checking active alarms” on page 396.

Table 10-36 Configuring Interface Alarms

Task	J-Web Configuration Editor	CLI Configuration Editor
Navigate to the Syslog level in the configuration hierarchy.	<ol style="list-style-type: none"> 1. In the J-Web interface, select CLI Tools → Point and Click CLI. 2. Next to Chassis, click Configure or Edit. 3. Next to Alarm, click Configure or Edit. See Figure 10-38 on page 395 	<p>From the [edit] hierarchy level, enter</p> <pre>{primary:node0}[edit] ibm@J58S-1> edit chassis alarm</pre>
Configure the system to generate a red interface alarm when a link down failure is detected on an Ethernet link.	<ol style="list-style-type: none"> 1. In the Ethernet field, click Configure. 2. From the Link down list, select red. 3. Click OK. See Figure 10-39 on page 396 	<p>Enter</p> <pre>{primary:node0}[edit chassis alarm] ibm@J58S-1> set ethernet link-down red</pre>

Task	J-Web Configuration Editor	CLI Configuration Editor
Configure the system to generate the following service alarms: Red alarm when Hw down is detected Red alarm when Linkdown is detected Red alarm when Pic hold reset is detected Yellow alarm when Pic reset is detected Yellow alarm when Rx errors is detected Red alarm when Sw down is detected Yellow alarm when Tx errors is detected	<ol style="list-style-type: none"> 1. In the Services field, click Configure. 2. From the Hw down list, select red. 3. From the Linkdown list, select red. 4. From the Pic hold reset list, select red. 5. From the Pic reset list, select yellow. 6. From the Rx errors list, select yellow. 7. From the Sw down list, select red. 8. From the Tx errors list, select yellow. 9. Click OK. See Figure 10-40 on page 396. 	Enter: <pre>{primary:node0}[edit chassis alarm] ibm@J58S-1> set services hw-down red linkdown red pic-hold-reset red pic-reset yellow rx-errors yellow sw-down red tx-errors yellow</pre>

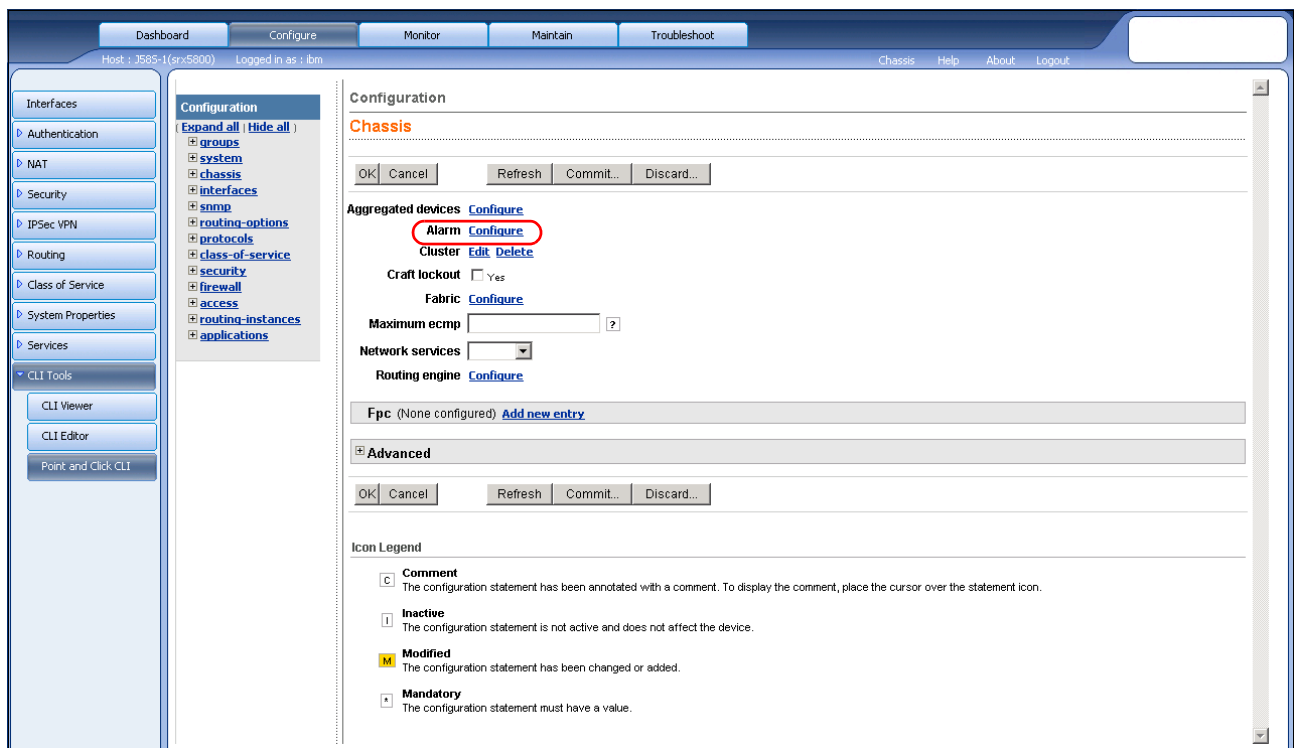


Figure 10-38 J-Web configure chassis alarms

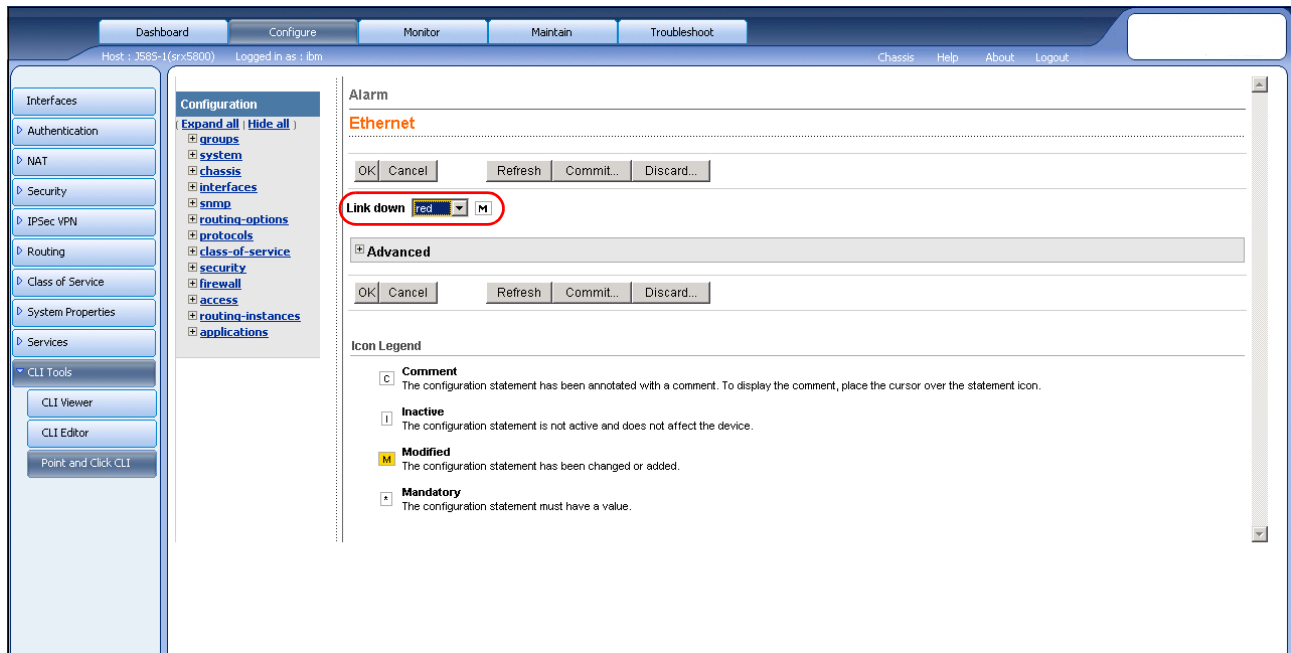


Figure 10-39 J-Web configure Ethernet alarms

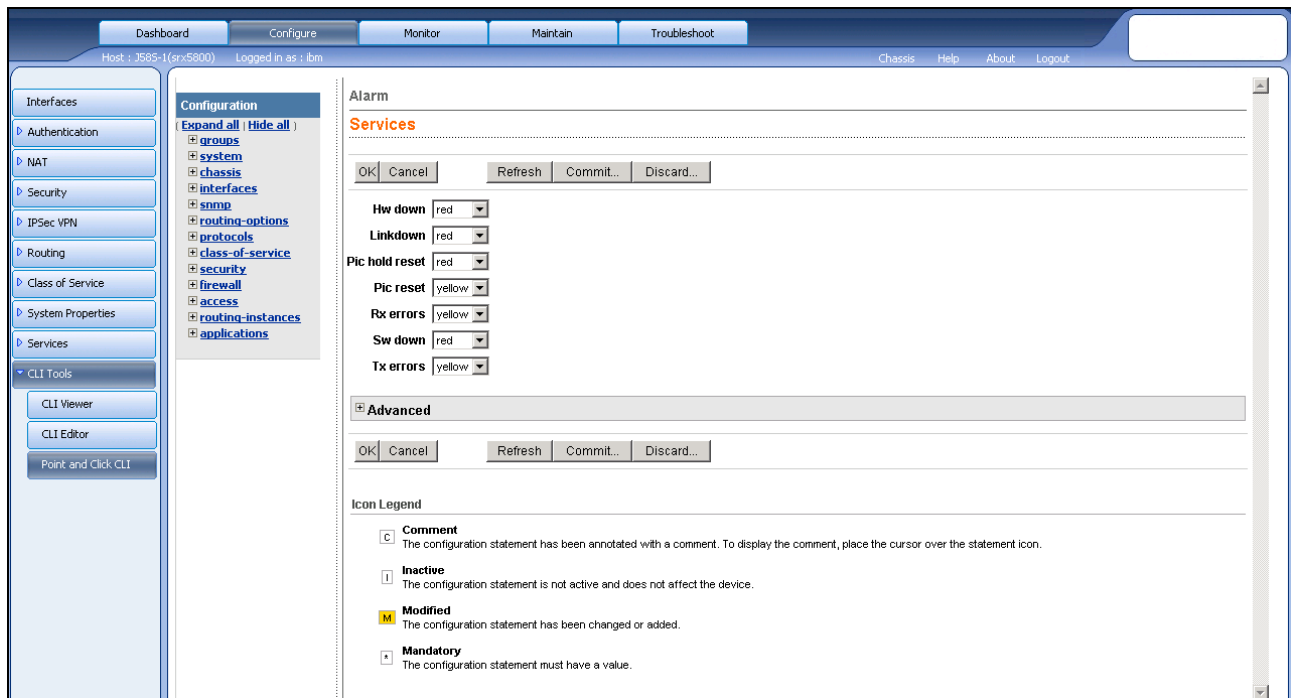


Figure 10-40 J-Web configure services alarms

10.4.4 Checking active alarms

To monitor alarms on the device, select **Monitor** → **Events and Alarms** → **View Alarms** in the J-Web user interface, see Figure 10-41 on page 397. The J-Web View Alarms page displays information about preset system and chassis alarms. For information about interface alarms, see “Interface alarm conditions” on page 390.

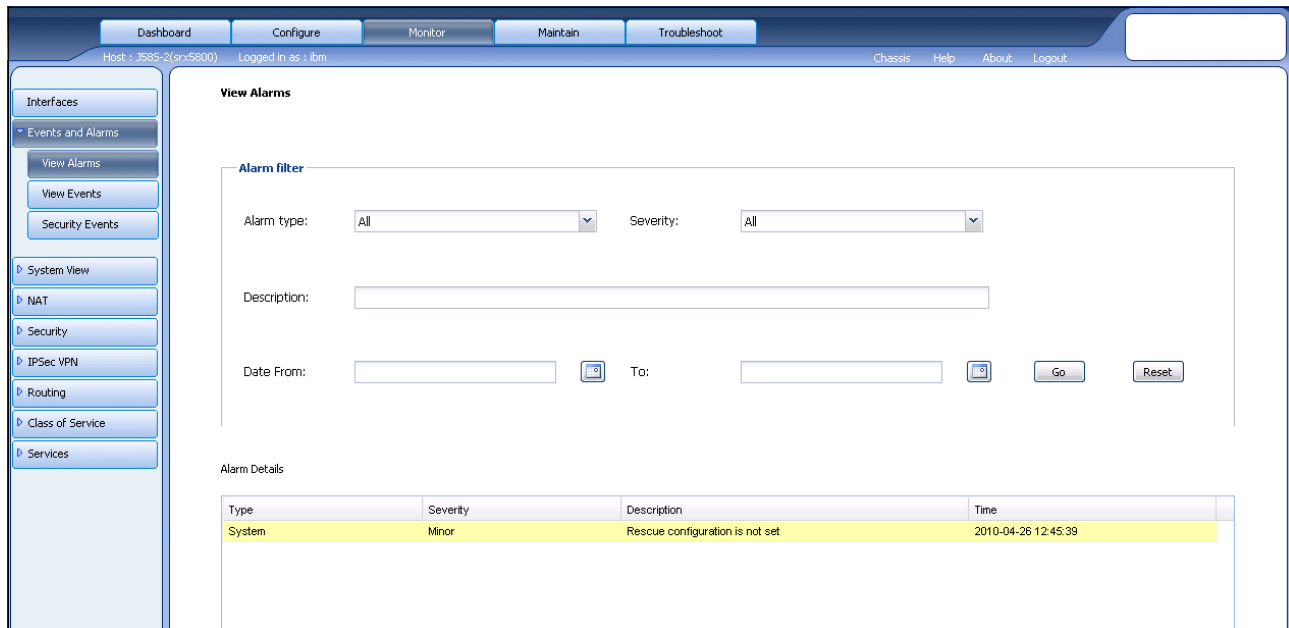


Figure 10-41 J-Web monitor alarms

Alternatively, you can enter the following show commands in the CLI editor:

show chassis alarms

show system alarms

The J-Web View Alarms page displays the following information about each alarm:

- ▶ **Type**—Type of alarm: System, Chassis, or All.
- ▶ **Severity**—Severity class of the alarm: Minor or Major.
- ▶ **Description**: Description of the alarm.
- ▶ **Time**: Time that the alarm was registered.

To filter which alarms are displayed, use the following options:

- ▶ **Alarm Type**: Specify which type of alarm to monitor: System, Chassis, or All. System alarms include FRU detection alarms (power supplies removed, for instance). Chassis alarms indicate environmental alarms such as temperature. For more information, see “Alarm types” on page 389.
- ▶ **Severity**: Specify the alarm severity that you want to monitor: Major, Minor, or All. A major (red) alarm condition requires immediate action. A minor (yellow) condition requires monitoring and maintenance. For more information, see “Alarm severity” on page 389.
- ▶ **Description**: Enter a brief synopsis of the alarms you want to monitor.
- ▶ **Date From**: Specify the beginning of the date range that you want to monitor. Set the date using the calendar pick tool.
- ▶ **To**: Specify the end of the date range that you want to monitor. Set the date using the calendar pick tool.
- ▶ **Go**: Executes the options that you specified.
- ▶ **Reset**: Clears the options that you specified.

Verifying the alarms configuration

To verify alarms configuration, select **Configure** → **CLI Tools** → **CLI Viewer** in the J-Web user interface, see Figure 10-42.

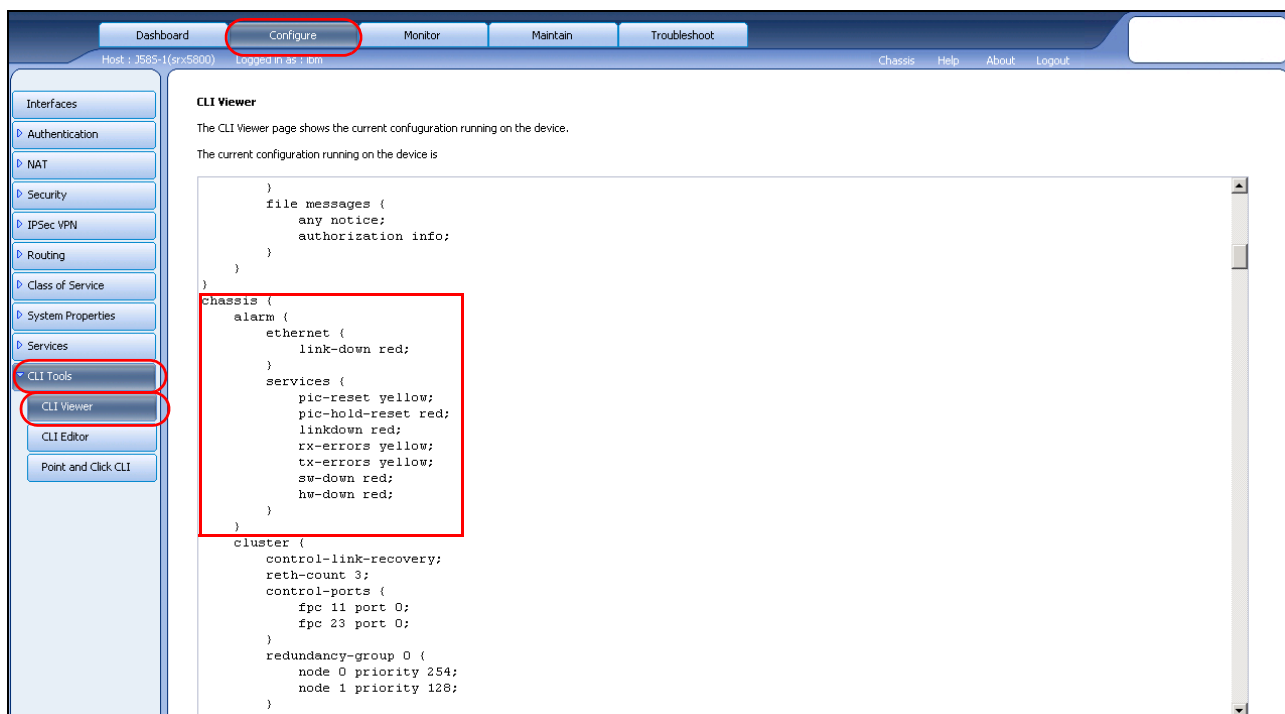


Figure 10-42 J-Web verify alarms configuration

Alternatively, from configuration mode in the CLI, enter the `show chassis alarms` command, as shown in Example 10-6.

Example 10-6 CLI verify alarms configuration

```
{primary:node0}[edit]
ibm@J58S-1# show chassis alarm
ethernet {
    link-down red;
}
services {
    pic-reset yellow;
    pic-hold-reset red;
    linkdown red;
    rx-errors yellow;
    tx-errors yellow;
    sw-down red;
    hw-down red;
}

{primary:node0}[edit]
ibm@J58S-1#
```

10.5 Additional material

For more information on all the topics discussed in this chapter, refer to the following documentation that can be found at the following website:

<http://www-947.ibm.com/systems/support/supportsite.wss/brandmain?brandind=5375876>

- ▶ *JUNOS Software System Basics and Services Command Reference GA32-0671-02.*
- ▶ *JUNOS Software Services Interfaces Configuration Guide GA32-0707-02*
- ▶ *JUNOS Software Network Management Configuration Guide GA32-0698-00*
- ▶ *JUNOS Software System Log Messages Reference GA32-0675-00*
- ▶ *JUNOS Software Network Interfaces Command Reference, GA32-0706-00*
- ▶ *JUNOS Software Routing Protocols and Policies Command Reference, GA32-0673-00*
- ▶ *JUNOS Software CLI User Guide, GA32-0697-00.*



Maintenance and analysis

In this chapter, we describe the most important tasks that are required for the maintenance and analysis of the IBM j-type s-series.

This chapter contains the following topics:

- ▶ Basic systems functions, such as rebooting, halting, managing files, backup configuration, and so on.
- ▶ Basic network utilities
- ▶ Diagnosing tools such as monitoring interface, traffic and system commands
- ▶ Managing Junos Software tasks such as upgrading and backing up
- ▶ Managing licenses
- ▶ File system overview
- ▶ Management of the configuration files
- ▶ Chassis and interfaces alarms

11.1 Basic systems functions

In this section, we explain how to perform basic systems tasks to:

- ▶ Reboot the system
- ▶ Halt the system
- ▶ Bring chassis components online and offline
- ▶ Restart software process
- ▶ Manage files
- ▶ Generate a rescue configuration
- ▶ Revert to rescue configuration
- ▶ Revert to default factory settings
- ▶ Recover root password

11.1.1 Rebooting or halting the system

To reboot the device using the CLI, execute the command in Example 11-1.

Example 11-1 System reboot

```
{primary:node0}
ibm@J58S-1> request system reboot ?
Possible completions:
  <[Enter]>      Execute this command
  at             Time at which to perform the operation
  in            Number of minutes to delay before operation
  media         Boot media for next boot
  message       Message to display to all users
  |            Pipe through a command
```

The following list contains the available options:

- ▶ **at time** (Optional): Time at which to reboot the software, which is specified in one of the following ways:
 - **now**: Stop or reboot the software immediately. This is the default.
 - **+minutes**: Number of minutes from now to reboot the software.
 - **yymmddhhmm**: Absolute time at which to reboot the software, which is specified as year, month, day, hour, and minute.
 - **hh:mm**: Absolute time on the current day at which to stop the software, which is specified in 24-hour time.
- ▶ **in minutes** (Optional): Number of minutes from now to reboot the software. This option is an alias for the **at +minutes** option.
- ▶ **media** (*compact-flash* | *disk* | *removable-compact-flash* | *usb*): (Optional) Boot medium for next boot.
- ▶ **message "text"**—(Optional): Message to display to all system users before stopping or rebooting the software.

Reboot requests are recorded in the system log files, which you can view with the **show log** command. Also, the names of any running processes that are scheduled to be shut down are changed. You can view the process names with the **show system processes** command.

To halt the device using the CLI, execute the command in Example 11-2.

Example 11-2 System halt

```
{primary:node0}
ibm@J58S-1> request system halt ?
Possible completions:
  <[Enter]>      Execute this command
  at             Time at which to perform the operation
  in            Number of minutes to delay before operation
  media         Boot media for next boot
  message       Message to display to all users
  other-routing-engine Halt other Routing Engine
  |            Pipe through a command
```

The following list contains the available options:

- ▶ **at time** (Optional): Time at which to halt the software, specified in one of the following ways:
 - **now**: Stop the software immediately. This is the default.
 - **+minutes**: Number of minutes from now to stop the software.
 - **yymmddhhmm**: Absolute time at which to stop the software, which is specified as year, month, day, hour, and minute.
 - **hh:mm**: Absolute time on the current day at which to stop the software, which is specified in 24-hour time.
- ▶ **in minutes** (Optional): Number of minutes from now to stop the software. This option is an alias for the **at +minutes** option.
- ▶ **media** (compact-flash | disk | removable-compact-flash | usb): (Optional) Boot medium for next boot.
- ▶ **message "text"**—(Optional): Message to display to all system users before stopping the software.

11.1.2 Bringing chassis components online and offline

You can use the CLI **request** commands to bring all chassis components (except Power Entry Modules and fans) online and offline, as shown in Example 11-3.

Example 11-3 Request chassis command

```
{primary:node0}
ibm@J58S-1> request chassis ?
Possible completions:
  cb             Change Control Board status
  cluster       Chassis cluster related requests
  fabric        Change fabric status
  fpc           Change Flexible PIC Concentrator status
  fpm           Change craft interface status
  pic          Change Physical Interface Card status
  routing-engine Change Routing Engine status
{primary:node0}
ibm@J58S-1> request chassis fpc ?
Possible completions:
  node          Change Flexible PIC Concentrator status of specific node
```

```

offline          Take FPC offline
online           Bring FPC online
restart          Restart FPC
slot             FPC slot number (0..23)
{primary:node0}
ibm@J58S-1> request chassis fpc offline ?
Possible completions:
  node           Change Flexible PIC Concentrator status of specific node
  slot           FPC slot number (0..23)
{primary:node0}
ibm@J58S-1> request chassis fpc offline slot ?
Possible completions:
<slot>          FPC slot number (0..23)

```

11.1.3 Restarting a software process

To correct an error condition, you might need to restart a software process that is running on the router. Use the **restart** command to force a restart of a software process.

Important: Restarting a software process during normal operation of a router can cause interruption of packet forwarding and loss of data.

Example 11-4 shows how to restart the routing process.

Example 11-4 Restarting the routing process

```

{primary:node0}
ibm@J58S-1> restart routing ?
Possible completions:
<[Enter]>       Execute this command
gracefully      Gracefully restart the process
immediately     Immediately restart (SIGKILL) the process
soft            Soft reset (SIGHUP) the process
|              Pipe through a command

```

The command presents the following options:

- ▶ Gracefully: Restarts the software process after performing clean-up tasks.
- ▶ Immediately: Restarts the software process without performing any clean-up tasks
- ▶ Soft: Rereads and reactivates the configuration without completely restarting the software processes, for example, BGP peers stay up and the routing table stays constant.

11.1.4 Managing files

You can use the J-Web interface or CLI to perform routine file management operations, such as archiving log files and deleting unused log files, cleaning up temporary files and crash files, and downloading log files from the routing platform to your computer.

You can also encrypt the configuration files with the CLI configuration editor to prevent unauthorized users from viewing sensitive configuration information. For an overview of the file system, see 11.6, “File system overview” on page 428.

The file management operations are:

► Cleanup

Use the CLI **request system storage cleanup** command to rotate log files and delete unnecessary files on the services router, as shown in Example 11-5. If you are running low on storage space, the file cleanup procedure quickly identifies files that can be deleted.

The file cleanup procedure performs the following tasks:

- Rotates log files: All information in the current log files is archived, old archives are deleted, and fresh log files are created.
- Deletes log files in **/var/log**: Any files that are not currently being written to are deleted.
- Deletes all crash files in **/var/crash**: Any core files that the device wrote during an error are deleted.
- Deletes all software images (*.tgz files) in **/var/sw/pkg**: Any software images copied to this directory during software upgrades are deleted.

Example 11-5 Requesting system cleanup

```
ibm@J58S-1> request system storage cleanup ?
Possible completions:
  <[Enter]>      Execute this command
  dry-run        Only list the cleanup candidates, do not remove them
  |              Pipe through a command
```

You can also use the **request system software delete-backup** command to delete the backup Junos software file (if it exists) to free up compact flash drive space. After running this command, you can no longer use the **request system software rollback** command to revert to the earlier version of the Junos software.

► Copying files

Use the **file copy** command to copy files from one place to another on the local router or between the local router and a remote system, as shown in Example 11-6.

Example 11-6 Copying files

```
ibm@J58S-1> file copy day1config ftp://ibm@10.1.1.203
Password for ibm@10.1.1.203:
ftp://ibm@10.1.1.203/day1config          100% of 299 B   16 kBps
```

► Deleting files

Use the **file delete** command to delete a file on the local router, as shown in Example 11-7.

Example 11-7 Deleting files

```
ibm@J58S-1> file delete day1config ?
Possible completions:
  <[Enter]>      Execute this command
  purge         Overwrite regular files before deleting them
  |              Pipe through a command
{master}
```

11.1.5 Setting rescue configuration

Using a rescue configuration, you can define a known working configuration or a configuration with a known state that you can roll back to at any time. This feature alleviates the necessity of having to remember the rollback number with the **rollback** command. Use the rescue configuration when you must roll back to a known configuration or as a last resort if your router configuration and the backup configuration files become damaged beyond repair.

To set the current active configuration as the rescue configuration:

```
ibm@J58S-1> request system configuration rescue save
```

To delete an existing rescue configuration:

```
ibm@J58S-1> request system configuration rescue delete
```

11.1.6 Reverting to rescue configuration

To return to the rescue configuration, use the **rollback rescue** configuration mode command:

```
ibm@J58S-1# rollback rescue
```

To activate the rescue configuration that you loaded, use the **commit** command:

```
ibm@J58S-1# commit
```

11.1.7 Reverting to default factory settings

If for any reason the current active configuration fails, you can revert to the default factory configuration. The default factory configuration contains the basic configuration settings. This is the first configuration of the device, and it is loaded when the device is first installed and powered on.

To load the default factory configuration:

1. Enter the **load factory-default** command:

```
ibm@J58S-1# load factory-default
```

2. Use the **set system root-authentication plain-text-password** command to set a new root password for the device:

```
ibm@J58S-1# set system root-authentication plain-text-password
```

3. Enter the root password, and enter it again for confirmation:

New password:

Retype new password:

Making the device remotely accessible: Before you commit changes, if you do not assign an IP address, create a local user account, and enter routing information, the device is no longer remotely accessible.

4. Use the **commit and-quit** command to commit the configuration and exit from configuration mode if the configuration contains no errors and the commit succeeds:

```
ibm@J58S-1# commit and-quit
```

5. Use the **request system reboot** command to reboot the device:

```
ibm@J58S-1> request system reboot
```

After the reboot, the factory default configuration is the running configuration.

11.1.8 Recovering passwords

The following steps describe the root password recovery procedure for the IBM j-type s-series:

1. Power off your device by unplugging the power cord or turning off the power at the wall switch.
2. Connect the console port to the management PC using the serial cable supplied with the device.
3. Power on your device by plugging in the power cord or turning on the power at the wall switch.
4. When the following prompt appears, press the Spacebar to access the switch's bootstrap loader command prompt:

Hit [Enter] to boot immediately, or space bar for command prompt. Booting [kernel] in 1 second...

5. At the following prompt, type **boot -s** to start the system in single-user mode:

loader> **boot -s**

6. At the following prompt, type **recovery** to start the root password recovery procedure:

Enter full pathname of shell or 'recovery' for root password recovery or RETURN for /bin/sh: **recovery**

A series of messages describe consistency checks, mounting of file systems, and initialization and checkout of management services. Next the CLI prompt appears.

7. Enter configuration mode in the CLI:

user@host> **configure**

8. Set the root password, for example:

user@host# **set system root-authentication plain-text-password**

9. At the following prompt, enter the new root password, for example:

New password: **ibm123**

Retype new password:

10. At the second prompt, reenter the new root password.

11. If you are finished configuring the network, commit the configuration.

root@host# **commit**

12. Exit configuration mode in the CLI.

root@host# **exit**

13. Exit operational mode in the CLI.

root@host> **exit**

14. At the prompt, enter y to reboot the switch.

Reboot the system? [y/n] **y**

11.2 Network utilities

In this section, we describe the basic network utilities that are commonly used for verification and troubleshooting.

11.2.1 Ping and traceroute

You can use the **ping** and **traceroute** commands to determine network reachability and to determine the path that packets take to reach a destination. In Example 11-8, we show the available options for these commands. Use Ctrl+c to stop ping and traceroute.

Example 11-8 Ping and Traceroute commands

```
ibm@J58S-1> ping 10.1.1.133 ?
Possible completions:
  <[Enter]>      Execute this command
  bypass-routing Bypass routing table, use specified interface
  count          Number of ping requests to send (1..2000000000 packets)
  detail         Display incoming interface of received packet
  do-not-fragment Don't fragment echo request packets (IPv4)
  inet           Force ping to IPv4 destination
  interface      Source interface (multicast, all-ones, unrouted packets)
  interval       Delay between ping requests (seconds)
  logical-system Name of logical system
+ loose-source   Intermediate loose source route entry (IPv4)
  no-resolve     Don't attempt to print addresses symbolically
  pattern        Hexadecimal fill pattern
  rapid          Send requests rapidly (default count of 5)
  record-route   Record and report packet's path (IPv4)
  routing-instance Routing instance for ping attempt
  size           Size of request packets (0..65468 bytes)
  source         Source address of echo request
  strict         Use strict source route option (IPv4)
+ strict-source  Intermediate strict source route entry (IPv4)
  tos            IP type-of-service value (0..255)
  ttl            IP time-to-live value (IPv6 hop-limit value) (1..255 hops)
  verbose        Display detailed output
  wait           Delay after sending last packet (seconds)
  |              Pipe through a command
{primary:node0}
ibm@J58S-1> ping 10.1.1.133
PING 10.1.1.133 (10.1.1.133): 56 data bytes
64 bytes from 10.1.1.133: icmp_seq=0 ttl=128 time=0.802 ms
64 bytes from 10.1.1.133: icmp_seq=1 ttl=128 time=0.235 ms
64 bytes from 10.1.1.133: icmp_seq=2 ttl=128 time=0.234 ms
^C
--- 10.1.1.133 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.234/0.424/0.802/0.268 ms

ibm@J58S-1> traceroute ?
Possible completions:
  <host>          Hostname or address of remote host
  as-number-lookup Look up AS numbers for each hop
```

bypass-routing	Bypass routing table, use specified interface
gateway	Address of router gateway to route through
inet	Force traceroute to IPv4 destination
interface	Name of interface to use for outgoing traffic
logical-system	Name of logical system
monitor	Monitor network connection to remote host
no-resolve	Don't attempt to print addresses symbolically
routing-instance	Name of routing instance for traceroute attempt
source	Source address to use in outgoing traceroute packets
tos	IP type-of-service field (IPv4) (0..255)
ttl	IP maximum time-to-live value (or IPv6 maximum hop-limit
value)	
wait	Number of seconds to wait for response (seconds)

```
{primary:node0}
ibm@J58S-1> traceroute 10.1.1.133
traceroute to 10.1.1.133 (10.1.1.133), 30 hops max, 40 byte packets
1  * * *
```

11.2.2 Telnet, SSH, and FTP clients

The CLI supports Telnet, SSH, and FTP clients. Example 11-9 shows the available arguments for these commands.

Example 11-9 Telnet, SSH, and FTP

```
ibm@J58S-1> telnet ?
Possible completions:
<host>      Hostname or address or remote host
8bit        Use 8-bit data path
bypass-routing  Bypass routing table, use specified interface
inet        Force telnet to IPv4 destination
interface   Name of interface for outgoing traffic
logical-system  Name of logical system
no-resolve   Don't attempt to print addresses symbolically
port        Port number or service name on remote host
routing-instance  Name of routing instance for telnet session
source      Source address to use in telnet connection
{primary:node0}
ibm@J58S-1> ssh ?
Possible completions:
<host>      Hostname or address of remote host
bypass-routing  Bypass routing table, use specified interface
inet        Force ssh to IPv4 destination
interface   Name of interface for outgoing traffic
logical-system  Name of logical system
routing-instance  Name of routing instance for ssh session
source      Source address for ssh session
v1          Force ssh to try protocol version 1 only
v2          Force ssh to try protocol version 2 only
{primary:node0}
ibm@J58S-1> ftp ?
Possible completions:
<host>      Hostname or address of remote host
bypass-routing  Bypass routing table, use specified interface
inet        Force FTP to IPv4 destination
```

inet6	Force FTP to IPv6 destination
interface	Name of interface for outgoing traffic
logical-system	Name of logical system
routing-instance	Name of routing instance for FTP session
source	Source address for FTP session

In a cluster environment, Fxp0 interface provides the management access, and it is limited to host traffic only. In this case, the traffic received through the fxp0 interface is not forwarded to any other interface in the system.

When its required to enable SSH, Telnet, and FTP services on specific interface, it must enabled under host inbound services configuration, as shown in Example 11-10.

Example 11-10 Telnet, SSH, and FTP under host inbound services

```
{primary:node0}[edit security zones security-zone trust]
ibm@J58S-1# show
host-inbound-traffic {
    system-services {
        ssh;
        telnet;
        ftp;
    }
    protocols {
        all;
    }
}
interfaces {
    reth0.0;
    ge-0/0/2.0;
    lo0.0;
}
```

11.3 Diagnosing tools

You can diagnose the device with CLI operational mode commands. CLI command output appears on the panel of your console or management device, or you can filter the output to a file.

In this section, we describe the following topics:

- ▶ Show interfaces command
- ▶ Monitor interface command
- ▶ Monitor traffic command
- ▶ Monitoring system commands
- ▶ Displaying log and traces

11.3.1 Show interfaces command

You can enter the following show commands in the CLI to view interface status and traffic statistics:

- ▶ **show interfaces terse**
- ▶ **show interfaces *interface-name***
- ▶ **show interfaces extensive *interface-name***

The **show interfaces terse** command displays summary information about interfaces, as shown in Example 11-11.

Example 11-11 Show interfaces terse command

```
ibm@J58S-1> show interfaces terse
```

Interface	Admin	Link	Proto	Local	Remote
ge-1/0/0	up	down			
ge-1/0/0.0	up	down	aenet	--> fab0.0	
ge-1/0/1	up	up			
ge-1/0/1.100	up	up	aenet	--> reth1.100	
ge-1/0/1.200	up	up	aenet	--> reth1.200	
ge-1/0/2	up	up			
ge-1/0/2.101	up	up	aenet	--> reth2.101	
ge-1/0/2.201	up	up	aenet	--> reth2.201	
ge-1/0/3	up	down			
ge-1/0/4	up	down			
ge-1/0/5	up	up			
ge-1/0/5.30	up	up	inet	192.168.30.1/24	
			multiservice		
ge-1/0/5.31	up	up	inet	192.168.31.1/24	
			multiservice		
ge-1/0/5.32767	up	up	multiservice		
ge-1/0/6	up	up			
ge-1/0/6.40	up	up	inet	192.168.40.1/24	
			multiservice		
ge-1/0/6.50	up	up	inet	192.168.50.1/24	
			multiservice		
ge-1/0/6.32767	up	up	multiservice		
ge-1/0/7	up	down			
ge-1/0/8	up	down			
ge-1/0/9	up	down			
ge-1/0/9.0	up	down	inet		
			multiservice		
mt-11/0/0	up	down			
dsc	up	up			
em0	up	up			
em0.0	up	up	inet	129.16.0.1/2	
			tnp	0x1100004	
em1	up	up			
em1.0	up	up	inet	129.16.0.1/2	
			tnp	0x1100004	
fab0	up	down			
fab0.0	up	down	inet	30.17.0.200/24	
fxp0	up	up			
fxp0.0	up	up	inet	10.1.1.12/24	
gre	up	up			
ipip	up	up			

```

irb                up    up
lo0                up    up
lo0.0              up    up    inet    10.8.8.2        --> 0/0
lo0.16384          up    up    inet    127.0.0.1       --> 0/0
lo0.16385          up    up    inet
lsi                up    up
mtun               up    up
pimd               up    up
pime               up    up
ppd0               up    up
ppe0               up    up
reth0              up    down
reth1              up    up
reth1.100          up    up    inet    192.168.100.1/24
                                multiservice
reth1.200          up    up    inet    192.168.200.1/24
                                multiservice
reth1.32767        up    down multiservice
reth2              up    up
reth2.101          up    up    inet    192.168.101.1/24
                                multiservice
reth2.201          up    up    inet    192.168.201.1/24
                                multiservice
reth2.32767        up    down multiservice
st0                up    up
st0.0              up    up    inet    192.168.20.10/24
tap                up    up

```

Table 11-1 shows the output fields of the command.

Table 11-1 Show interfaces terse command output fields

Field name	Description
Interface	Interface name.
Admin	Whether the interface is turned on (up) or off (down).
Link	Link state: up or down.
Proto	Protocol family configured on the logical interface. A logical interface on a router that supports Ethernet OAM always shows the multiservice protocol.
Local	Local IP address of the logical interface.
Remote	Remote IP address of the logical interface.

The **show interfaces interface-name** command displays standard information about the specified interface, as shown in Example 11-12.

Example 11-12 Show interfaces command

```

ibm@J58S-1> show interfaces ge-1/0/0
Physical interface: ge-1/0/0, Enabled, Physical link is Down
  Interface index: 133, SNMP ifIndex: 506
  Link-level type: 64, MTU: 9014, Speed: 1000mbps, BPDU Error: None,

```



```

MAC-REWRITE Error: None, Loopback: Disabled, Source filtering: Disabled,
Flow control: Enabled, Auto-negotiation: Enabled, Remote fault: Online
Device flags   : Present Running Down
Interface flags: Hardware-Down SNMP-Traps Internal: 0x4000
Link flags     : None
CoS queues     : 8 supported, 4 maximum usable queues
Schedulers    : 0
Current address: 00:1f:12:fc:5f:f1, Hardware address: 00:1f:12:fc:58:a5
Last flapped   : 2010-05-24 18:25:51 UTC (19:44:39 ago)
Input rate     : 0 bps (0 pps)
Output rate    : 0 bps (0 pps)
Active alarms  : LINK
Active defects : LINK

```

```

Logical interface ge-1/0/0.0 (Index 70) (SNMP ifIndex 546)
  Flags: Device-Down SNMP-Traps Encapsulation: ENET2
  Input packets : 0
  Output packets: 1492793
  Security: Zone: Null
  Protocol aenet, AE bundle: fab0.0   Link Index: 0

```

The **show interfaces extensive** command displays all possible information about the interfaces installed in the device. You can specify a particular interface, as shown in Example 11-13

Example 11-13 Show interface extensive command

```

ibm@J58S-1> show interfaces extensive ge-1/0/1.100
Logical interface ge-1/0/1.100 (Index 74) (SNMP ifIndex 609) (Generation 160)
  Flags: SNMP-Traps 0x4000 VLAN-Tag [ 0x8100.100 ] Encapsulation: ENET2
  Traffic statistics:
    Input bytes   : 644313246
    Output bytes  : 639039720
    Input packets : 828807
    Output packets: 755836
  Local statistics:
    Input bytes   : 0
    Output bytes  : 517910
    Input packets : 0
    Output packets: 5753
  Transit statistics:
    Input bytes   : 0                               544 bps
    Output bytes  : 0                               0 bps
    Input packets : 0                               1 pps
    Output packets: 0                               0 pps
  Security: Zone: Null
  Flow Statistics :
  Flow Input statistics :
    Self packets : 0
    ICMP packets : 0
    VPN packets  : 0
    Multicast packets : 0
    Bytes permitted by policy : 0
    Connections established : 0
  Flow Output statistics:

```

```

Multicast packets : 0
Bytes permitted by policy : 0
Flow error statistics (Packets dropped due to):
Address spoofing: 0
Authentication failed: 0
Incoming NAT errors: 0
Invalid zone received packet: 0
Multiple user authentications: 0
Multiple incoming NAT: 0
No parent for a gate: 0
No one interested in self packets: 0
No minor session: 0
No more sessions: 0
No NAT gate: 0
No route present: 0
No SA for incoming SPI: 0
No tunnel found: 0
No session for a gate: 0
No zone or NULL zone binding 0
Policy denied: 0
Security association not active: 0
TCP sequence number out of window: 0
Syn-attack protection: 0
User authentication errors: 0
Protocol aenet, AE bundle: reth1.100 Link Index: 0, Generation: 181, Route
table: 0

```

Table 11-2 shows the output fields.

Table 11-2 Show interface extensive command output fields

Name	Description
Traffic statistics	Number of packets and bytes transmitted and received on the physical interface.
Local statistics	Number of packets and bytes transmitted and received on the physical interface.
Transit statistics	Number of packets and bytes transiting the physical interface.
Flow input statistics	Statistics about packets received by flow module.
Flow output statistics	Statistics about packets sent by flow module.
Flow error statistics	Statistics about errors in the flow module.

11.3.2 Monitor interface command

Use the CLI **monitor interface** command to display real-time traffic, error, alarm, and filter statistics about a physical or logical interface, as shown in Example 11-14.

Example 11-14 Monitoring interface

```

{primary:node0}
ibm@J58S-1> monitor interface ge-1/0/0

```

```

J58S-1                               Seconds: 10                               Time: 14:24:06
                                          Delay: 6/0/6

Interface: ge-1/0/0, Enabled, Link is Up
Encapsulation: Fabric-Member-Ethernet, Speed: 1000mbps
Traffic statistics:                               Current delta
  Input bytes:                               0 (0 bps)                               [0]
  Output bytes:                             329116116 (10384 bps)                       [12980]
  Input packets:                             0 (0 pps)                               [0]
  Output packets:                           256781 (1 pps)                           [10]
Error statistics:
  Input errors:                               0                               [0]
  Input drops:                               0                               [0]
  Input framing errors:                       0                               [0]
  Policed discards:                           0                               [0]
  L3 incompletes:                             0                               [0]
  L2 channel errors:                           0                               [0]
  L2 mismatch timeouts:                       0                               [0]
  Carrier transitions:                         9                               [0]
  Output errors:                               0                               [0]
  Output drops:                               0                               [0]
  Aged packets:                               0                               [0]
Active alarms : None
Active defects: None
Input MAC/Filter statistics:
  Unicast packets                             253514                               [10]
  Broadcast packets                           0                               [0]
  Multicast packets                           0                               [0]
  Oversized frames                           0                               [0]
  Packet reject count                         0                               [0]
  DA rejects                                 0                               [0]
  SA rejects                                 0                               [0]
Output MAC/Filter Statistics:
  Unicast packets                             256784                               [10]
  Broadcast packets                           0                               [0]
  Multicast packets                           0                               [0]
  Packet pad count                           0                               [0]
  Packet error count                         0                               [0]

```

```

Next='n', Quit='q' or ESC, Freeze='f', Thaw='t', Clear='c', Interface='i'

```

The real-time statistics are updated every second. The Current delta and Delta columns display the amount the statistics counters have changed since the monitor interface command was entered or since you cleared the delta counters.

Table 11-3 lists the keys that you use to control the display.

Table 11-3 Monitor interface options

Key	Description
n	Displays information about the next interface. The device scrolls through the physical and logical interfaces in the same order in which they are displayed by the show interfaces terse command.

Key	Description
q or ESC	Quits the command and returns to the command prompt.
f	Freezes the display, halting the update of the statistics and delta counters.
t	Thaws the display, resuming the update of the statistics and delta counters.
c	Clears (returns to 0) the delta counters in the Current delta column. The statistics counters are not cleared.
i	Displays information about a different interface. You are prompted for the name of a specific interface.

11.3.3 Monitor traffic command

Use the CLI **monitor traffic** command, shown in Example 11-15, to display packet headers that are transmitted through network interfaces.

Preserving system performance: Using the monitor traffic command can degrade system performance. Use filtering options, such as count and matching, to minimize the impact to packet throughput on the system.

Example 11-15 Monitor traffic command

```
ibm@J58S-1> monitor traffic ?
Possible completions:
  <[Enter]>      Execute this command
  absolute-sequence  Display absolute TCP sequence numbers
  brief           Display brief output
  count           Number of packets to receive (0..1000000 packets)
  detail          Display detailed output
  extensive        Display extensive output
  interface        Name of interface
  layer2-headers   Display link-level header on each dump line
  matching          Expression for headers of receive packets to match
  no-domain-names  Don't display domain portion of hostnames
  no-promiscuous   Don't put interface into promiscuous mode
  no-resolve        Don't attempt to print addresses symbolically
  no-timestamp      Don't print timestamp on each dump line
  print-ascii      Display packets in ASCII when displaying in hexadecimal
  format
  print-hex         Display packets in hexadecimal format
  resolve-timeout   Period of time to wait for each name resolution
  (1..4294967295 seconds)
  size              Amount of each packet to receive (bytes)
  |                 Pipe through a command
```

To limit the packet header information displayed by the **monitor traffic** command, include the matching *expression* option.

For more information about matching options, refer to page 341 in the *Junos Software Administration Guide*:

<http://www.juniper.net/techpubs/software/junos-security/junos-security10.1/junos-security-admin-guide/junos-security-admin-guide.pdf>

Example 11-16 shows how to use the **monitor traffic** command to match TCP traffic.

Example 11-16 Monitoring traffic

```
ibm@J58S-1> monitor traffic matching tcp
verbose output suppressed, use <detail> or <extensive> for full protocol decode
Address resolution is ON. Use <no-resolve> to avoid any reverse lookup delay.
Address resolution timeout is 4s.
Listening on fxp0, capture size 96 bytes
```

Reverse lookup for 10.1.1.12 failed (check DNS reachability).
Other reverse lookup failures will not be reported.
Use <no-resolve> to avoid reverse lookups on IP addresses.

```
15:33:26.541703 In IP 10.1.1.203.3793 > 10.1.1.12.telnet: . ack 1322464508 win
15463
15:33:26.541786 Out IP truncated-ip - 221 bytes missing! 10.1.1.12.telnet >
10.1.1.203.3793: P 1:242(241) ack 0 win 65535
15:33:26.742888 In IP 10.1.1.203.3793 > 10.1.1.12.telnet: . ack 242 win 15222
15:33:27.375293 Out IP truncated-ip - 158 bytes missing! 10.1.1.12.telnet >
10.1.1.203.3793: P 242:420(178) ack 0 win 65535
15:33:27.547631 In IP 10.1.1.203.3793 > 10.1.1.12.telnet: . ack 420 win 15044
15:33:27.547708 Out IP truncated-ip - 270 bytes missing! 10.1.1.12.telnet >
10.1.1.203.3793: P 420:710(290) ack 0 win 65535
15:33:27.748800 In IP 10.1.1.203.3793 > 10.1.1.12.telnet: . ack 710 win 16384
15:33:28.379360 Out IP truncated-ip - 390 bytes missing! 10.1.1.12.telnet >
10.1.1.203.3793: P 710:1120(410) ack 0 win 65535
15:33:28.554496 In IP 10.1.1.203.3793 > 10.1.1.12.telnet: . ack 1120 win 15974
15:33:29.379047 Out IP truncated-ip - 187 bytes missing! 10.1.1.12.telnet >
10.1.1.203.3793: P 1120:1327(207) ack 0 win 65535
15:33:29.559474 In IP 10.1.1.203.3793 > 10.1.1.12.telnet: . ack 1327 win 15767
15:33:30.378891 Out IP truncated-ip - 188 bytes missing! 10.1.1.12.telnet >
10.1.1.203.3793: P 1327:1535(208) ack 0 win 65535
15:33:30.565372 In IP 10.1.1.203.3793 > 10.1.1.12.telnet: . ack 1535 win 15559
15:33:31.378728 Out IP truncated-ip - 188 bytes missing! 10.1.1.12.telnet >
10.1.1.203.3793: P 1535:1743(208) ack 0 win 65535
15:33:31.571299 In IP 10.1.1.203.3793 > 10.1.1.12.telnet: . ack 1743 win 15351
^C
21 packets received by filter
0 packets dropped by kernel
```

Example 11-17 shows how to use the **monitor interface traffic** command to look at the incremental input output packets and pps rate on each interface.

Example 11-17 Monitor interface traffic

```
ibm@J58S-1> monitor interface traffic
```

```
Bytes=b, Clear=c, Delta=d, Packets=p, Quit=q or ESC, Rate=r, Up=^U, Down=^D
J58S-1                               Seconds: 2                               Time: 15:14:09
```

Interface	Link	Input packets	(pps)	Output packets	(pps)
ge-0/0/0	Up	0	(0)	3774537	(2)
ge-0/0/1	Up	0	(0)	0	(0)
ge-0/0/2	Up	1109560	(1)	127284	(0)
ge-0/0/3	Down	0	(0)	0	(0)
ge-0/0/4	Down	0	(0)	0	(0)
ge-0/0/5	Down	0	(0)	0	(0)
ge-0/0/6	Down	0	(0)	0	(0)
ge-0/0/7	Up	578866	(0)	4719	(0)
ge-0/0/8	Down	0	(0)	0	(0)
ge-0/0/9	Down	0	(0)	0	(0)
ge-0/0/10	Down	0	(0)	0	(0)
ge-0/0/11	Down	0	(0)	0	(0)
ge-1/0/0	Up	1109493	(1)	127330	(0)
ge-1/0/1	Up	0	(0)	0	(0)
ge-1/0/2	Down	0	(0)	0	(0)
ge-1/0/3	Down	0	(0)	0	(0)
ge-1/0/4	Down	0	(0)	0	(0)
ge-1/0/5	Down	0	(0)	0	(0)
ge-1/0/6	Down	0	(0)	0	(0)

Bytes=b, Clear=c, Delta=d, Packets=p, Quit=q or ESC, Rate=r, Up=^U, Down=^D

11.3.4 Monitoring system commands

You can view system properties by entering the following show commands in the CLI configuration editor, as shown in Example 11-18:

- ▶ Show system uptime: Shows time since the system and processes started.
- ▶ Show system user: Shows users who are currently logged in.
- ▶ Show system storage: Shows local storage data.
- ▶ Show version: Shows software process revision levels.
- ▶ Show chassis hardware: Shows installed hardware components.

Example 11-18 Show system commands

```
ibm@J58S-1> show system uptime
node0:
-----
Current time: 2010-05-24 19:39:15 UTC
System booted: 2010-05-18 15:09:34 UTC (6d 04:29 ago)
Protocols started: 2010-05-18 15:25:00 UTC (6d 04:14 ago)
Last configured: 2010-05-24 19:32:22 UTC (00:06:53 ago) by ibm
7:39PM up 6 days, 4:30, 7 users, load averages: 0.01, 0.00, 0.00

{primary:node0}
ibm@J58S-1> show system users
node0:
-----
7:39PM up 6 days, 4:30, 7 users, load averages: 0.01, 0.00, 0.00
USER      TTY      FROM          LOGIN@  IDLE WHAT
ibm       p0       10.1.1.8      Sat02PM 2days -cli (cli)
ibm       p1       10.1.1.203    1:17PM   - -cli (cli)
```

```

ibm      p2      10.1.1.202      1:43PM  5:54 -cli (cli)
ibm      p3      10.1.1.202      1:48PM  1:02 -cli (cli)
ibm      p4      10.1.1.200      3:55PM  28 -cli (cli)
ibm      p5      10.1.1.8        5:31PM  1:36 -cli (cli)
ibm      p6      10.1.1.202      7:23PM  6 -cli (cli)
ibm      jweb1   10.1.1.202      5:34PM  2:04
ibm      jweb2   10.1.1.8        6:08PM  1:31

```

```

{primary:node0}
ibm@J58S-1> show system storage
node0:

```

```

-----
Filesystem      Size      Used      Avail  Capacity  Mounted on
/dev/ad0s1a     885M      364M      450M    45%      /
devfs           1.0K      1.0K      0B      100%     /dev
/dev/md0        426M      426M      0B      100%     /junos
/cf             885M      364M      450M    45%     /junos/cf
devfs           1.0K      1.0K      0B      100%     /junos/dev/
procfs          4.0K      4.0K      0B      100%     /proc
/dev/ad0s1e     98M       36K       90M     0%       /config
/dev/ad2s1f     34G       206M      31G     1%       /var
/dev/md1        1007M     522K      926M    0%       /mfs
/var/jail       34G       206M      31G     1%       /jail/var
/var/log        34G       206M      31G     1%       /jail/var/log
devfs           1.0K      1.0K      0B      100%     /jail/dev

```

```

{primary:node0}
ibm@J58S-1> show version
node0:

```

```

-----
Hostname: J58S-1
Model: srx5800
JUNOS Software Release [10.1R2.8]

```

```

{primary:node0}
ibm@J58S-1> show chassis hardware
node0:

```

```

-----
Hardware inventory:
Item          Version  Part number  Serial number  Description
Chassis                               JN1146C0EAGA  SRX 5800
Midplane      REV 01   710-024803   TS2134         SRX 5800 Backplane
FPM Board     REV 01   710-024632   XJ6267         Front Panel Display
PDM           Rev 03   740-013110   QCS1249500C    Power Distribution Module
PEM 0         Rev 03   740-023514   QCS1248E010    PS 1.7kW; 200-240VAC in
PEM 1         Rev 03   740-023514   QCS1250E00N    PS 1.7kW; 200-240VAC in
PEM 2         Rev 03   740-023514   QCS1250E028    PS 1.7kW; 200-240VAC in
Routing Engine 0 REV 03   740-023530   9009020052     RE-S-1300
CB 0          REV 03   710-024802   XG7819         SRX5k SCB
CB 1          REV 03   710-024802   XA9198         SRX5k SCB
FPC 1         REV 13   750-020235   WX3745         SRX5k DPC 40x 1GE
  CPU         REV 02   710-024633   WX3596         SRX5k DPC PMB
  PIC 0                               BUILTIN        BUILTIN        10x 1GE RichQ
  Xcvr 0      REV 01   740-011782   PAR1805        SFP-SX

```

Xcvr 1	REV 01	740-011613	PCE029W	SFP-SX
Xcvr 2	REV 02	740-011613	PG12NM6	SFP-SX
Xcvr 5	REV 02	740-011613	PG12C0Z	SFP-SX
Xcvr 6	REV 02	740-011613	PG12BG0	SFP-SX
PIC 1		BUILTIN	BUILTIN	10x 1GE RichQ
Xcvr 0	REV 02	740-011613	PG12RPL	SFP-SX
Xcvr 1	REV 02	740-011613	PG12MZJ	SFP-SX
PIC 2		BUILTIN	BUILTIN	10x 1GE RichQ
Xcvr 0	REV 02	740-011613	PG12RKG	SFP-SX
PIC 3		BUILTIN	BUILTIN	10x 1GE RichQ
FPC 11	REV 09	750-023996	XJ4894	SRX5k SPC
CPU	REV 02	710-024633	XG8708	SRX5k DPC PMB
PIC 0		BUILTIN	BUILTIN	SPU Cp-Flow
PIC 1		BUILTIN	BUILTIN	SPU Flow
Fan Tray 0	REV 05	740-014971	VS0008	Fan Tray
Fan Tray 1	REV 05	740-014971	VS0061	Fan Tray

11.3.5 Displaying log and trace files

You can enter the **monitor start** command to display real-time additions to system logs and trace files:

```
ibm@J58S-1> monitor start filename
```

When the device adds a record to the file specified by *filename*, the record is displayed on the panel, for example, if you configured a system log file named **system-log** (by including the **syslog** statement at the [edit system] hierarchy level), you can enter the **monitor start system-log** command to display the records added to the system log.

To display a list of files that are being monitored, enter the **monitor list** command. To stop the display of records for a specified file, enter the **monitor stop filename** command.

For more information about configuring syslog, refer to Chapter 10, “Management and monitoring” on page 327.

11.4 Managing Junos software

IBM j-type s-series are delivered with Junos software preinstalled. When you power on the device, it starts (boots) up using its primary boot device. These devices also support secondary boot devices allowing you to back up your primary boot device and configuration.

As new features and software fixes become available, you can upgrade your software to use them. Before an upgrade, we recommend that you back up your primary boot device.

You can configure the primary or secondary boot device with a “snapshot” of the current configuration, default factory configuration, or rescue configuration. You can also replicate the configuration for use on another device, or configure a boot device to receive core dumps for troubleshooting.

If the IBM j-type s-series device does not have a secondary boot device configured and the primary boot device becomes corrupted, you can reload the Junos recovery software package onto the corrupted CompactFlash card with either a UNIX or Microsoft Windows computer.

11.4.1 Understanding software packaging

A software package name is in the following format:

package-name-m.nZx.y-domestic-signed.tgz

Where:

- ▶ package-name is the name of the package, for example: jinstall-ex-4200.
- ▶ m.n is the software release, with m representing the major release number and n representing the minor release number, for example: 9.5.
- ▶ Z indicates the type of software release, where R indicates released software and B indicates beta-level software.
- ▶ x.y represents the version of the major software release (x) and an internal tracking number (y), for example: 1.6.
- ▶ For most Junos packages, *domestic-signed* is used for the United States and Canada and *export-signed* for worldwide distribution.

An example of Junos software package is:

jinstall-ex-4200-9.5R1.6-domestic-signed.tgz

11.4.2 Understanding recovery software packaging

Download a recovery software package, also known as an install media package, to recover a primary CompactFlash card.

A recovery software package name is in the following format:

package-name-m.nZx-export-cfnnn.gz

Where:

- ▶ package-name is the name of the package, for example: junos-jsr.
- ▶ m.n is the software release, with m representing the major release number, for example: 8.5.
- ▶ Z indicates the type of software release. For example, R indicates released software, and B indicates beta-level software.
- ▶ x.y represents the software build number and spin number, for example: 1.1.
- ▶ export indicates that the recovery software package is the exported worldwide software package version.
- ▶ cfnnn indicates the size of the target CompactFlash card in megabytes, for example, cf256.

11.4.3 System snapshot

You can create copies of the software using the system snapshot feature. Use the **request system snapshot** CLI command shown in Example 11-19 to create a boot device on an alternate medium, to replace the primary boot device or serve as a backup.

Example 11-19 System snapshot

```
ibm@J58S-1> request system snapshot ?
```

Possible completions:

<[Enter]>	Execute this command
media	Media to snapshot to
node	Archive data and executable areas of specific node
partition	Partition the media
re0	Archive data and executable areas of RE0
re1	Archive data and executable areas of RE1
routing-engine	Archive data and executable areas of specific routing engine
	Pipe through a command

11.4.4 Upgrading Junos software

Typically, you upgrade your device software by downloading a software image to your device from another system on your local network. Using the J-Web interface or the CLI to upgrade, the device downloads the software image, decompresses the image, and installs the decompressed software. Finally, you reboot the device, at which time it boots from the upgraded software.

Before an upgrade, back up your primary boot device onto a secondary storage device. If you have a power failure during an upgrade, the primary boot device can fail or become corrupted. In either case, if a backup device is not available, the device might be unable to boot and come back online. Creating a backup also stores your active configuration files and log files and ensures that you recover to a known, stable environment in case of an unsuccessful upgrade.

During a successful upgrade, the upgrade package completely reinstalls the existing software. It retains configuration files, log files, and similar information from the previous version.

Upgrading Junos software: This procedure applies only to upgrading one Junos software (heritage services) release to another or upgrading one Junos software release to another. To upgrade Junos software (heritage services) to Junos software, see the *Junos Software Migration Guide*.

You can use the following procedure to upgrade the Junos software on an IBM j-type s-series.

1. Before installing the software upgrade, verify the available space on the CompactFlash card. See Example 11-17 on page 417.
2. Download the software package.
3. (Optional) Back up the current software configuration as described in 11.4.3, “System snapshot” on page 421.
4. (Optional) Copy the software package to the device. You can use FTP to copy the file to the /var/tmp directory.

This step is optional because the Junos Software can also be upgraded when the software image is stored at a remote location.

5. Install the new package on the device:

```
user@switch> request system software add unlink no-copy source
```

Replace *source* with one of the following paths:

- For a software package that is installed from a local directory on the device:
/pathname/package-name.
- For a software package that is downloaded and installed from a remote location:
 - ftp://hostname/pathname/package-name
 - http://hostname/pathname/package-name

By default, the **request system software add** command uses the **validate** option to validate the software package against the current configuration as a prerequisite to adding the software package. This validation ensures that the device can reboot successfully after the software package is installed. This is the default behavior when you are adding a software package.

The **unlink** option removes the package at the earliest opportunity so that the device has enough storage capacity to complete the installation.

The **no-copy** option specifies that a software package is installed, but a copy of the package is not saved. Include this option if you do not have enough space on the CompactFlash card to perform an upgrade that keeps a copy of the package on the device.

6. After the software package is installed, reboot the device:

```
user@host> request system reboot
```

When the reboot is complete, the device displays the login prompt.

7. After the reboot has completed, log in and verify that the new version of the software is properly installed:

```
user@switch> show version
```

8. Remember to back up the new current configuration to the secondary device as described in step 2.

In the Example 11-20, we show a software upgrade on IBM j-type s-series.

Example 11-20 Software upgrade on IBM j-type s-series

```
root@J58S-1> show version
Hostname: J58S-1
Model: srx5800
JUNOS Software Release [10.1R1.8]

root@J58S-1> request system software add unlink no-copy
//var/tmp/junos-srx5000-10.1R2.8-domestic.tgz
NOTICE: Validating configuration against junos-srx5000-10.1R2.8-domestic.tgz.
NOTICE: Use the 'no-validate' option to skip this if desired.
Checking compatibility with configuration
Initializing...
Verified manifest signed by PackageProduction_10_1_0
Using //var/tmp/junos-srx5000-10.1R2.8-domestic.tgz
Checking junos requirements on /
Available space: 645572 require: 189314
Saving boot file package in /var/sw/pkg/junos-boot-srx5000-10.1R2.8.tgz
Verified manifest signed by PackageProduction_10_1_0
Hardware Database regeneration succeeded
Validating against /config/juniper.conf.gz
cp: /var/validate/chroot/var/etc/resolv.conf and /etc/resolv.conf are identical
(not copied).
cp: /var/validate/chroot/var/etc/hosts and /etc/hosts are identical (not copied).
```

```
mgd: commit complete
Validation succeeded
Installing package '//var/tmp/junos-srx5000-10.1R2.8-domestic.tgz' ...
Verified SHA1 checksum of issu-indb.tgz
Verified junos-boot-srx5000-10.1R2.8.tgz signed by PackageProduction_10_1_0
Verified junos-srx5000-10.1R2.8-domestic signed by PackageProduction_10_1_0
Available space: 645572 require: 189314
Saving boot file package in /var/sw/pkg/junos-boot-srx5000-10.1R2.8.tgz
JUNOS 10.1R2.8 will become active at next reboot
WARNING: A reboot is required to load this software correctly
WARNING:      Use the 'request system reboot' command
WARNING:      when software installation is complete
Saving state for rollback ...
Removing //var/tmp/junos-srx5000-10.1R2.8-domestic.tgz

root@J58S-1> request system reboot
Reboot the system ? [yes,no] (no) yes
```

```
*** FINAL System shutdown message from root@J58S-1 ***
System going down IMMEDIATELY
```

```
Shutdown NOW!
```

11.4.5 Downgrading Junos software

When you upgrade your software, the device creates a backup image of the software that was previously installed and installs the requested software upgrade.

To downgrade the software, you can use the backup image of the software that was previously installed, which is saved on the device. If you revert to the previous image, this backup image is used, and the image of the running software is deleted. You can downgrade to only the software release that was installed on the device before the current release with this method.

You can revert to the previous version of software using the **request system software rollback** command in the CLI. For the changes to take effect, you must reboot the device. To downgrade to an earlier version of software, follow the procedure for upgrading, using the software image labeled with the appropriate release.

Note: This procedure applies only to downgrading one Junos software release to another or one Junos software services release to another. To downgrade Junos software services to the Junos software, see the *Junos Software Migration Guide*.

To downgrade software with the CLI follow the next steps:

1. Enter the **request system software rollback** command to return to the previous Junos Software version:

```
ibm@J58S-1> request system software rollback
```

2. Reboot the device:

```
ibm@J58S-1> request system reboot
```

11.5 Managing licences

To enable some Junos software features, you must purchase, install, and manage separate software licenses. For those features that require a license, the presence on the device of the appropriate software license keys (passwords) determines whether you can use the feature.

For features that require a license, you must install and properly configure the license to use the feature. Although the device allows you to commit a configuration that specifies a feature requiring a license when the license is not present, you are prohibited from actually using the feature.

Successful commitment of a configuration does not imply that the required licenses are installed. If a required license is not present, the system provides a warning message after it commits the configuration rather than failing to commit it because of a license violation.

Note: Each feature license is tied to exactly one software feature, and that license is valid for exactly one device. IBM j-type s-series only require licences for IDP Update Signature feature.

11.5.1 Licence key components

A license key consists of two parts:

- ▶ License ID: Alphanumeric string that uniquely identifies the license key. When a license is generated, it is given a license ID.
- ▶ License data: Block of binary data that defines and stores all license key objects.

11.5.2 Generating a license key

To generate a license key:

1. Gather the authorization code that you received when you purchased your license and your device serial number.
2. Go to the Juniper Networks licensing page at:
<https://www.juniper.net/1crs/generateLicense.do>
3. Enter the device serial number and authorization code in the Web page and click **Generate**. Depending on the type of license you purchased, you will receive one of the following:
 - a. License key: If you purchased a perpetual license, you will receive a license key from the licensing management system. You can enter this key directly into the system to activate the feature on your device.
 - b. License key entitlement: If you purchased a subscription-based license, you will receive a license key entitlement from the licensing management system. You can use this entitlement to validate your license on the Juniper Networks licensing server and download the feature license from the server to your device.

11.5.3 Managing Junos software licenses

To manage Junos software licenses with the CLI, perform the following tasks:

- ▶ Adding New Licenses

- ▶ Deleting a License
- ▶ Updating New Licenses
- ▶ Saving License Keys

Adding new licences

To add a new license key to the device with the CLI:

1. Enter operational mode in the CLI.
2. Enter one of the following CLI commands:
 - To add a license key from a file or URL, enter the following command, specifying the filename or the URL where the key is located:


```
ibm@J58S-1> request system license add ?
```

 Possible completions:

<code><filename></code>	Filename (URL, local, remote, or floppy)
-------------------------------	--

 Use the filename option to activate a perpetual license directly on the device. (Most feature licenses are perpetual.) Use the url option to send a subscription-based license key entitlement (such as UTM) to the Juniper Networks licensing server for authorization. If authorized, the server downloads the license to the device and activates it.
 - To add a license key from the terminal, enter the following command:


```
ibm@J58S-1> request system license add terminal
```
3. When prompted, enter the license key, separating multiple license keys with a blank line. If the license key you enter is invalid, an error is generated when you press Ctrl+D to exit license entry mode.

Deleting a license

To delete a license key from the device with the CLI:

1. Enter operational mode in the CLI.
2. Enter the following command for each license, specifying the license ID. You can delete only one license at a time.


```
ibm@J58S-1> request system license delete license-name
```

Updating new licenses

To update a license key from the device with the CLI:

1. Enter operational mode in the CLI.
2. Enter one of the following CLI commands:
 - a. To automatically update the license keys, enter the following command:


```
ibm@J58S-1> request system license update
```

 The request system license update command will always use the default Juniper license server <https://ae1.juniper.net>
 You can only use this command to update subscription-based licenses (such as UTM).
 - b. To automatically update the trial license keys, enter the following command:


```
ibm@J58S-1> request system license update trial
```

Saving licenses keys

To save the licenses installed on the device to a file with the CLI:

1. Enter operational mode in the CLI.
2. To save the installed license keys to a file or URL, enter the following command:

```
ibm@J58S-1> request system license save ?
```

Possible completions:

```
<filename>          Filename (URL, local, remote, or floppy)
```

11.5.4 Verifying Junos software licenses

To verify license management, perform the following tasks:

- ▶ Displaying installed licenses
- ▶ Displaying license usage
- ▶ Displaying installed license keys

Displaying installed licenses

Issue the command shown in Example 11-21 to verify that the expected licenses are installed and active on the device.

Example 11-21 Showing licenses

```
ibm@J58S-1> show system license
```

License usage:

Feature name	Licenses used	Licenses installed	Licenses needed	Expiry
idp-sig	0	1	0	2010-06-14
00:00:00 UTC				

Licenses installed:

```
License identifier: JUNOS253192
License version: 2
Valid for device: JN1146C0EAGA
Features:
  idp-sig          - IDP Signature
                  date-based, 2010-04-15 00:00:00 UTC - 2010-06-14 00:00:00 UTC
```

Displaying licenses usage

Issue the command shown in Example 11-22 to verify that the licenses fully cover the feature configuration on the device.

Example 11-22 Showing system license usage

```
ibm@J58S-1> show system license usage
```

Feature name	Licenses used	Licenses installed	Licenses needed	Expiry
idp-sig	0	1	0	2010-06-14
00:00:00 UTC				

Displaying installed licence keys

Type the command shown in Example 11-23 to verify the license keys installed on the device.

Example 11-23 Showing system license keys

```
ibm@J58S-1> show system license keys
```

```
JUNOS253192 aeaqea qmjhd cmjugz btarkb i5aqqb qcdw4x
             vonwa4 udasjg m75zew s7br66 r5pofa 26felh
             y2njxb elgkf5 yur14p om361q 6vcqfk 5zjz5o
             7yy
```

11.6 File system overview

The following are the most important directories in the Junos software file system:

- ▶ `/`: The root file system located on the boot device (usually the primary compact-flash drive).
- ▶ `/config`: Located on the boot device, contains the current operational configuration, the last three committed configurations and the rescue configuration.
- ▶ `/config/db/config`: Contains up to 46 additional previous committed configurations.
- ▶ `/var`: Contains the following subdirectories.
 - `/var/home`: Contains users home directories, created when you create user accounts.
 - `/var/log`: Contains system logs and tracing files.
 - `/var/tmp`: Contains daemon core files and temporary files.

Important: Junos software deletes the contents of the `/var` directory during software upgrades, so it is very important to back up required files and logs externally.

You can view the device's directory structure and individual files by issuing the **file** command in operational mode as shown in Example 11-24. Use the list option to see the directory structure of the router.

The default directory for the **file list** command is the home directory of the user logged in to the router. In fact, the user's home directory is the default directory for most of Junos software commands requiring a filename.

Example 11-24 File command

```
ibm@J58S-1> file ?
Possible completions:
  <[Enter]>      Execute this command
  archive        Archives files from the system
  checksum       Calculate file checksum
  compare        Compare files
  copy           Copy files (local or remote)
  delete         Delete files from the system
  list           List file information
  rename         Rename files
  show           Show file contents
  source-address Local address to use in originating the connection
  |             Pipe through a command

{primary:node0}
ibm@J58S-1> file list
```



```
/var/home/ibm/:  
.ssh/
```

To view the contents of other file directories, specify the directory location, as shown in Example 11-25.

Example 11-25 File list command

```
ibm@J58S-1> file list /config
```

```
/config:  
.snap/  
juniper.conf.1.gz  
juniper.conf.2.gz  
juniper.conf.3.gz  
juniper.conf.gz  
juniper.conf.md5  
juniper.conf.spu.gz  
license/  
rescue.conf.gz  
usage.db
```

We show, in Example 11-26, how to use the router's context-sensitive help system to locate a directory.

Example 11-26 Locating a directory

```
ibm@J58S-1> file list /?
```

Possible completions:

<[Enter]>	Execute this command
<path>	Path to list
/COPYRIGHT	Size: 6187, Last changed: May 11 01:42:22
/a/	Last changed: May 11 06:00:20
/altconfig/	Last changed: May 11 06:00:20
/altroot/	Last changed: May 11 06:00:20
/b/	Last changed: May 11 06:00:20
/bin/	Last changed: May 11 06:00:20
/boot/	Last changed: May 17 16:56:20
/c/	Last changed: May 11 06:00:20
/cf/	Last changed: May 17 16:56:29
/config/	Last changed: May 24 15:09:45
/d/	Last changed: May 11 06:00:20
/data/	Last changed: May 11 06:00:20
/dev/	Last changed: May 18 15:09:40
/e/	Last changed: May 11 06:00:20
/etc/	Last changed: May 11 06:00:20
/f/	Last changed: May 11 06:00:20
/g/	Last changed: May 11 06:00:20
/jail/	Last changed: May 11 06:00:20
/kernel	Size: 189341128, Last changed: May 11 06:01:47
/kernel.old	Size: 188919699, Last changed: Apr 15 18:15:30
/libexec/	Last changed: May 11 06:00:20
/mfs/	Last changed: May 18 15:10:01
/mnt/	Last changed: May 11 06:00:20
/modules/	Last changed: May 11 06:00:20
/mount.post	Size: 2049, Last changed: May 11 01:42:48

/opt/	Last changed: Feb 12 18:49:43
/packages/	Last changed: May 17 16:59:03
/pkg/	Last changed: May 11 06:00:20
/proc/	Last changed: May 24 15:58:06
/root/	Last changed: May 17 16:56:29
/sbin/	Last changed: May 11 06:00:20
/staging/	Last changed: May 24 15:30:30
/tmp/	Last changed: May 24 15:30:30
/umount.pre	Size: 434, Last changed: May 11 01:42:48
/usr/	Last changed: May 11 06:00:20
/var/	Last changed: May 18 15:10:25

To display the contents of a file issue the command shown in Example 11-27.

Example 11-27 File show command

```
ibm@J58S-1> file show /var/log/inventory
Apr 15 18:20:32 CHASSISD release 10.1R1.8 built by builder on 2010-02-12 16:33:28
UTC
Apr 15 18:20:33 Midplane - part number 710-024803, serial number TS2134
Apr 15 18:20:34 CB - part number 710-024802, serial number XG7819
Apr 15 18:20:35 srx5800 chassis, serial number JN1146C0EAGA
Apr 15 18:20:35 FPC 11 - part number 750-023996, serial number XJ4894
Apr 15 18:20:36 CB 0 - part number 710-024802, serial number XG7819
Apr 15 18:20:37 CB 1 - part number 710-024802, serial number XA9198
Apr 15 18:20:38 Routing Engine 0 - part number 740-023530, serial number
9009020052
```

11.7 Configuration management

In this section, we describe the most important tasks needed to manage the configuration in Junos software.

11.7.1 Saving the configuration

When you edit a configuration, you work in a copy of the current configuration to create a candidate configuration. The changes you make to the candidate configuration are visible in the CLI immediately, so if multiple users are editing the configuration at the same time, all users can see all changes.

To have a candidate configuration take effect, you commit the changes. At this point, the candidate file is checked for proper syntax, activated, and marked as the current, operational software configuration file. If multiple users are editing the configuration, when you commit the candidate configuration, all changes made by all the users take effect.

In addition to saving the current configuration, the CLI saves the current operational version and the previous 49 versions of committed configurations. The most recently committed configuration is version 0 (the current operational version, which is the default configuration that the system returns to if you roll back to a previous configuration), and the oldest saved configuration is version 49.

The currently operational Junos Software configuration is stored in the file `juniper.conf`, and the last three committed configurations are stored in the files `juniper.conf.1`, `juniper.conf.2`,

and juniper.conf.3. These four files are located in the directory /config, which is on the router's flash drive. The remaining 46 previous versions of committed configurations, the files juniper.conf.4 through juniper.conf.49, are stored in the directory /var/db/config on the hard disk.

11.7.2 Returning to the most recently committed configuration

To return to the most recently committed configuration and load it into configuration mode without activating it, use the **rollback** configuration mode command:

```
ibm@J58S-1# rollback
```

To activate the configuration to which you rolled back, use the **commit** command:

```
ibm@J58S-1# commit
```

11.7.3 Returning to a previously committed configuration

To return to a configuration prior to the most recently committed one, include the number in the **rollback** command. The most recently saved configuration is number 0 (which is the default configuration to which the system returns), and the oldest saved configuration is number 49:

```
ibm@J58S-1# rollback 3
```

11.7.4 Displaying previous configurations

To display previous configurations, including the rollback number, date, time, the name of the user who committed changes, and the method of commit, use the **rollback ?** command as shown in Example 11-28.

Example 11-28 Rollback options

```
ibm@J58S-1# rollback ?
Possible completions:
  <[Enter]>      Execute this command
  0              2010-05-24 15:09:45 UTC by ibm via cli
  1              2010-05-22 13:53:19 UTC by ibm via cli
  2              2010-05-22 03:47:07 UTC by ibm via cli
  3              2010-05-22 03:34:59 UTC by ibm via cli
  4              2010-05-22 03:29:40 UTC by ibm via cli
  5              2010-05-22 03:25:54 UTC by ibm via cli
  6              2010-05-22 03:08:17 UTC by ibm via cli
  7              2010-05-22 00:38:48 UTC by ibm via cli
  8              2010-05-22 00:16:28 UTC by ibm via cli
  9              2010-05-22 00:12:03 UTC by ibm via cli
  10             2010-05-21 23:38:19 UTC by ibm via cli
  11             2010-05-21 23:37:20 UTC by ibm via cli
  12             2010-05-21 16:36:29 UTC by ibm via cli
  13             2010-05-21 16:15:14 UTC by ibm via cli
  14             2010-05-21 15:59:49 UTC by ibm via cli
  15             2010-05-21 15:27:33 UTC by ibm via cli
  16             2010-05-21 15:18:21 UTC by ibm via cli
  17             2010-05-21 15:15:55 UTC by ibm via cli
  18             2010-05-21 14:57:30 UTC by ibm via cli
  19             2010-05-21 14:17:43 UTC by ibm via cli
```

```

20          2010-05-21 14:10:25 UTC by ibm via cli
21          2010-05-21 14:08:44 UTC by ibm via cli
22          2010-05-21 04:20:04 UTC by ibm via cli
23          2010-05-21 04:15:31 UTC by ibm via cli
24          2010-05-21 04:05:00 UTC by ibm via cli
25          2010-05-21 02:55:49 UTC by ibm via cli
26          2010-05-21 02:55:22 UTC by ibm via cli
27          2010-05-21 02:12:32 UTC by ibm via cli
28          2010-05-21 01:49:46 UTC by ibm via cli
29          2010-05-21 01:37:13 UTC by ibm via cli
30          2010-05-21 01:33:53 UTC by ibm via cli
31          2010-05-21 00:20:56 UTC by ibm via cli
32          2010-05-21 00:17:08 UTC by ibm via cli
33          2010-05-21 00:11:41 UTC by ibm via cli
34          2010-05-21 00:08:23 UTC by ibm via cli
35          2010-05-20 21:37:12 UTC by ibm via cli
36          2010-05-20 21:36:34 UTC by ibm via cli
37          2010-05-20 21:31:05 UTC by ibm via cli
38          2010-05-20 21:28:22 UTC by ibm via cli
39          2010-05-20 21:26:17 UTC by ibm via cli
40          2010-05-20 21:23:05 UTC by ibm via cli
41          2010-05-20 21:21:41 UTC by ibm via cli
42          2010-05-20 21:19:12 UTC by ibm via cli
43          2010-05-20 20:50:14 UTC by ibm via junoscript
44          2010-05-20 20:48:55 UTC by ibm via junoscript commit
synchronize
45          2010-05-20 20:23:53 UTC by ibm via junoscript
46          2010-05-20 20:22:32 UTC by ibm via junoscript
47          2010-05-20 20:21:24 UTC by ibm via cli
48          2010-05-20 20:15:01 UTC by ibm via cli
49          2010-05-20 19:18:51 UTC by ibm via cli
rescue     2010-05-20 20:06:39 UTC by ibm via cli
|          Pipe through a command

```

11.7.5 Comparing configuration changes

In configuration mode only, when you have made changes to the configuration and want to compare the candidate configuration with a prior version, you can use the **compare** command to display the configuration. The **compare** command compares the candidate configuration with either the current committed configuration or a configuration file and displays the differences between the two configurations. To compare configurations, specify the **compare** command after the pipe:

```
ibm@J58S-1# show | compare (filename | rollback n)
```

Filename is the full path to a configuration file. The file must be in the proper format: a hierarchy of statements.

n is the index into the list of previously committed configurations. The most recently saved configuration is number 0, and the oldest saved configuration is number 49. If you do not specify arguments, the candidate configuration is compared against the active configuration file (/config/juniper.conf).

The comparison output uses the following conventions:

- Statements that are only in the candidate configuration are prefixed with a plus sign (+).

- Statements that are only in the comparison file are prefixed with a minus sign (-).
- Statements that are unchanged are prefixed with a single blank space ().

The Example 11-29 shows the comparison between the active configuration and rollback 3 configuration.

Example 11-29 Configuration comparison

```
{primary:node0}[edit]
ibm@J58S-1# show | compare rollback 1
[edit interfaces lo0 unit 0 family inet]
+       filter {
+           input protect-re;
+       }
[edit firewall family inet]
+       filter CoS-TEST { ... }
+       filter protect-re {
+           term permit-ssh {
+               from {
+                   source-address {
+                       10.1.1.0/24;
+                   }
+                   protocol tcp;
+                   destination-port [ ssh telnet ];
+               }
+           then accept;
+       }
+ }
```

11.7.6 Saving a configuration to a file

You might want to save the configuration to a file so that you can edit it with a text editor of your choice. You can save your current configuration to an ASCII file, which saves the configuration in its current form, including any uncommitted changes. If more than one user is modifying the configuration, all changes made by all users are saved.

To save software configuration changes to an ASCII file, use the **save** configuration mode command:

```
ibm@J58S-1# save day1config
Wrote 77 lines of configuration to 'day1config'
```

The contents of the current level of the statement hierarchy are saved, along with the statement hierarchy containing it. This allows a section of the configuration to be saved, while fully specifying the statement hierarchy.

By default, the configuration is saved to a file in your home directory.

When you issue this command from anywhere in the hierarchy (except the top level), a replace tag is automatically included at the beginning of the file. You can use the replace tag to control how a configuration is loaded from a file.

11.7.7 Loading a configuration from a file

You can create a file, copy the file to the local router, and then load the file into the CLI. After you load the file, you can commit it to activate the configuration on the router, or you can edit the configuration interactively using the CLI and commit it at a later time.

You can also create a configuration while typing at the terminal and then load it. Loading a configuration from the terminal is generally useful when you are cutting existing portions of the configuration and pasting them elsewhere in the configuration.

To load an existing configuration file that is located on the router, use the **load** configuration mode command. In the Example 11-30 we show the different options available with this command.

Example 11-30 Load command

```
ibm@J58S-1# load ?
Possible completions:
  factory-default  Override existing configuration with factory default
  merge            Merge contents with existing configuration
  override         Override existing configuration
  patch            Load patch file into configuration
  replace          Replace configuration data
  set              Execute set of commands on existing configuration
  update           Update existing configuration

ibm@J58S-1# load override ?
Possible completions:
  <filename>      Filename (URL, local, remote, or floppy)
  h               Size: 1781, Last changed: May 21 14:55:09
  soon-may-21     Size: 13736, Last changed: May 21 04:59:31
  terminal         Use login terminal
```

To load a configuration from the terminal, specify the **terminal** option. Type ^D (CTRL+D) to end input.

To replace an entire configuration, specify the **override** option at any level of the hierarchy. An override operation discards the current candidate configuration and loads the configuration in filename or the one that you type at the terminal. When you use the override option and commit the configuration, all system processes reparse the configuration.

To replace portions of a configuration, specify the **replace** option. For this operation to work, you must include replace: tags in the file or configuration you type at the terminal. The software searches for the replace: tags, deletes the existing statements of the same name, if any, and replaces them with the incoming configuration. If there is no existing statement of the same name, the replace operation adds to the configuration the statements marked with the replace: tag.

To replace only the configuration that has changed, specify the **update** option at any level of the hierarchy. An update operation compares the current configuration and the current candidate configuration, and loads only the changes between these configurations in filename or the one that you type at the terminal. When you use the update operation and commit the configuration, Junos software attempts to notify the smallest set of system processes that are affected by the configuration change.

To combine the current configuration and the configuration in filename or the one that you type at the terminal, specify the **merge** option. A merge operation is useful when you are adding a new section to an existing configuration. If the existing configuration and the incoming configuration contain conflicting statements, the statements in the incoming configuration override those in the existing configuration.

To change part of the configuration with a patch file and mark only those parts as changed, specify the **patch** option.

For more information about this topic refer to GA32-0697-02, *JUNOS Software CLI User Guide*:

http://www-01.ibm.com/support/docview.wss?rs=1331&context=SGPMK5&dc=DA400&uid=isg3T7000171&loc=en_US&cs=utf-8&lang=en

11.8 Chassis and interfaces alarms

In this section, we describe the different classes of chassis and interfaces alarms for the IBM j-type s-series family. For more information about configuring alarms refer to Chapter 10, “Management and monitoring” on page 327.

When the Routing Engine detects an alarm condition, it lights the red or yellow alarm LED on the craft interface as appropriate. To view a more detailed description of the alarm cause, issue the **show chassis alarms** command:

```
ibm@J58S-1> show chassis alarms
node0:
```

```
-----
No alarms currently active
```

```
node1:
```

```
-----
No alarms currently active
```

There are two classes of alarm messages:

- ▶ Chassis alarms: Indicate a problem with a chassis component such as the cooling system or power supplies.
- ▶ Interface alarms: Indicate a problem with a specific network interface.

11.8.1 Interface LEDs on J34S and J36S

LEDs on the IBM j-type s-series Ethernet Appliances J34S and J36S display the status of various components:

- ▶ Common form-factor module (CFM) LED: One LED labeled OK/FAIL on the faceplate of each IOC, SPC, and NPC indicates the CFM's status.
- ▶ SFB LEDs: Ten LEDs, labeled ALARM (2 LEDs), SFB, HA, CFM SERVICE, CFM OK/FAIL, RE0, RE1, PWR, and FAN, on the SFB faceplate indicate the status of the SFB. If no LEDs are lit, the Routing Engine may still be booting or the SFB is not receiving power.
- ▶ Routing Engine LEDs: Five LEDs, labeled MASTER, HDD, STATUS for the Routing Engine function, STATUS for the PFE controller, and OK/FAIL for general board status, on

the Routing Engine faceplate indicate the status of the Routing Engine and hard disk drive.

- ▶ SRX Clustering Module (SCM) LED: One LED, labeled OK/FAIL on the SCM faceplate indicates the status of the SCM.
- ▶ Power supply LEDs: One LED on each power supply faceplate indicates the status of that power supply.

11.8.2 Alarm relay contacts on J56S and J58S

The craft interface of IBM j-type s-series Ethernet Appliances J56S and J58S has two alarm relay contacts for connecting the system to external alarm devices. Whenever a system condition triggers either the red or yellow alarm on the craft interface, the alarm relay contacts are also activated. The alarm relay contacts are located on the upper right of the craft interface.

11.8.3 Craft Interface LEDs on J56S and J58S

The craft interface of IBM j-type s-series Ethernet Appliances J56S and J58S is the panel on the front of the device located above the card cage that contains LEDs and buttons that allow you to troubleshoot the device.

LEDs on the craft interface include the following:

- ▶ Alarm LEDs: One large red circular LED and one large yellow triangular LED, located on the upper right of the craft interface, indicate two levels of alarm conditions. The circular red LED lights to indicate a critical condition that can result in a system shutdown. The triangular yellow LED lights to indicate a less severe condition that requires monitoring or maintenance. Both LEDs can be lit simultaneously. A condition that causes an alarm LED to light also activates the corresponding alarm relay contact on the craft interface.
- ▶ Host subsystem LEDs: Three LEDs, MASTER, ONLINE, and OFFLINE, indicate the status of the host subsystem. A green MASTER LED indicates that the host is functioning as the master. The ONLINE LED indicates that the host is online. The OFFLINE LED indicates that the host is installed but the routing engine is offline. The host subsystem LEDs are located on the left of the craft interface and are labeled RE0 and RE1.
- ▶ Power supply LEDs: Two LEDs (PEM) indicate the status of each power supply. Green indicates that the power supply is functioning normally. Red indicates that the power supply is not functioning normally. The power supply LEDs are located in the center craft interface, and are labeled 0 through 3.
- ▶ IOC and SPC LEDs: Two LEDs, OK and FAIL, indicate the status of each IOC or SPC. Green indicates OK and red indicates a failure. The IOC and SPC LEDs are located along the bottom of the craft interface.
- ▶ SCB LEDs: Two LEDs, OK and FAIL, indicate the status of each SCB. Green indicates OK and red indicates a failure. The SCB LEDs are located in the center of the craft interface along the bottom, and are labeled 0 and 1.
- ▶ Fan LEDs: Two LEDs indicate the status of the fan. Green indicates OK and red indicates FAIL. The fan LEDs are located on the upper left of the craft interface.

11.8.4 Component LEDs on J56S and J58S

The following LEDs are located on various device components and display the status of those components on IBM j-type s-series Ethernet Appliances J56S and J58S:

- ▶ IOC LED: One LED labeled OK/FAIL on each IOC faceplate indicates the IOC's status.
- ▶ SPC LED: One LED labeled OK/FAIL on each SPC faceplate indicates the SPC's status.
- ▶ Port module LED: One LED labeled OK/FAIL on each port module faceplate indicates the port module's status.
- ▶ SCB LEDs: Three LEDs, labeled FABRIC ACTIVE, FABRIC ONLY, and OK/FAIL, on each SCB faceplate indicate the status of the SCB. If no LEDs are lit, the master RE may still be booting or the SCB is not receiving power.
- ▶ RE LEDs: Four LEDs, labeled MASTER, HDD, ONLINE, and FAIL on the Routing Engine faceplate indicate the status of the Routing Engine and hard disk drive.
- ▶ Power supply LEDs: One LED on each power supply faceplate indicates the status of that power supply.

11.9 More information

For more information about IBM j-type s-series maintenance and analysis refer to *JUNOS Software Administration Guide*:

<http://www.juniper.net/techpubs/software/junos-security/junos-security10.1/junos-security-admin-guide/junos-security-admin-guide.pdf>

For more information about Junos CLI commands refer to *JUNOS Software CLI Reference*:

<http://www.juniper.net/techpubs/software/junos-security/junos-security10.1/junos-security-cli-reference/junos-security-cli-reference.pdf>

For more information about troubleshooting hardware components on IBM j-type s-series Ethernet Appliance J34S refer to *SRX3400 Services Gateway Hardware Guide*:

http://www.juniper.net/techpubs/en_US/release-independent/junos/information-products/topic-collections/hardware/srx-series/srx3400/book-srx3400-hw-ig.pdf

For more information about troubleshooting hardware components on IBM j-type s-series Ethernet Appliance J36S refer to *SRX3600 Services Gateway Hardware Guide*:

http://www.juniper.net/techpubs/en_US/release-independent/junos/information-products/topic-collections/hardware/srx-series/srx3600/book-srx3600-hw-ig.pdf

For more information about troubleshooting hardware components on IBM j-type s-series Ethernet Appliance J56S refer to *SRX5600 Services Gateway Hardware Guide*:

http://www.juniper.net/techpubs/en_US/release-independent/junos/information-products/topic-collections/hardware/srx-series/srx5600/book-SRX5600-hw-ig.pdf

For more information about troubleshooting hardware components on IBM j-type s-series Ethernet Appliance J58S refer to *SRX5800 Services Gateway Hardware Guide*:

http://www.juniper.net/techpubs/en_US/release-independent/junos/information-products/topic-collections/hardware/srx-series/srx5800/book-srx5800-hw-ig.pdf

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

For information about ordering these publications, see “How to get Redbooks” on page 441. Note that some of the documents referenced here may be available in softcopy only.

- ▶ *IBM j-type Data Center Networking Introduction*, SG24-7820
- ▶ *IBM j-type Ethernet Switches and Routers Implementation*, SG24-7882

Other publications

These publications are also relevant as further information sources.

IBM j-type Ethernet Appliance the related hardware documentation:

- ▶ *IBM Ethernet Appliance J34S Getting Started Guide*, GA32-0749
- ▶ *IBM Ethernet Appliance J36S Getting Started Guide*, GA32-0751
- ▶ *IBM Ethernet Appliance J56S Getting Started Guide*, GA32-0753
- ▶ *IBM Ethernet Appliance J58S Getting Started Guide*, GA32-0755

IBM j-type Ethernet switches the related hardware documentation:

- ▶ *IBM Ethernet Switch J48E Complete Hardware Guide*, GA32-0663
- ▶ *IBM Ethernet Switch J48E Quick Start*, GA32-0664
- ▶ *IBM Ethernet Switch J08E Complete Hardware Guide*, GA32-0665
- ▶ *IBM Ethernet Switch J08E Quick Start*, GA32-0666
- ▶ *IBM Ethernet Switch J16E Complete Hardware Guide*, GA32-0667
- ▶ *IBM Ethernet Switch J16E Quick Start*, GA32-0668

IBM j-type Ethernet routers the related hardware documentation:

- ▶ *IBM j-type m-series Ethernet Router Dense Port Concentrators (DPC) Guide*, GA32-0661
- ▶ *IBM j-type m-series Ethernet Routing Engine Installation Instructions*, GA32-0682
- ▶ *IBM j-type m-series Ethernet Services PIC Guide*, GA32-0662
- ▶ *IBM Ethernet Router J02M Hardware Guide*, GA32-0655
- ▶ *IBM Ethernet Router J02M Quick Start*, GA32-0656
- ▶ *IBM Ethernet Router J06M Hardware Guide*, GA32-0657
- ▶ *IBM Ethernet Router J06M Quick Start*, GA32-0658
- ▶ *IBM Ethernet Router J11M Hardware Guide*, GA32-0659
- ▶ *IBM Ethernet Router J11M Quick Start*, GA32-0660
- ▶ *JUNOS Software IBM j-type m-series Ethernet Routers Solutions Guide*, GA32-0683

JUNOS software documentation:

- ▶ *Juniper Web Device Manager for IBM j-type Ethernet Switches and Routers Interface User Guide*, GA32-0688
- ▶ *JUNOS Software Access Privilege Configuration Guide*, GA32-0696

- ▶ *JUNOS Software Broadband Subscriber Management Solutions Guide*, GA32-0709
- ▶ *JUNOS Software Class of Service Configuration Guide*, GA32-0738
- ▶ *JUNOS Software CLI User Guide*, GA32-0697
- ▶ *JUNOS Software Configuration and Diagnostic Automation Guide*, GA32-0679
- ▶ *JUNOS Software Ethernet Routing Engine Media Upgrade Kit*, GA32-0681
- ▶ *JUNOS Software Feature Guide*, GA32-0739
- ▶ *JUNOS Software Hierarchy and RFC Reference*, GA32-0712
- ▶ *JUNOS Software High Availability Configuration Guide*, GA32-0670
- ▶ *JUNOS Software IBM j-type m-series Ethernet Routers Layer 2 Configuration Guide*, GA32-0708
- ▶ *JUNOS Software Installation and Upgrade Guide*, GA32-0695
- ▶ *JUNOS Software Interfaces Command Reference*, GA32-0672
- ▶ *JUNOS Software JUNOScript API Guide*, GA32-0674
- ▶ *JUNOS Software MPLS Applications Configuration Guide*, GA32-0702
- ▶ *JUNOS Software Multicast Protocols Configuration Guide*, GA32-0703
- ▶ *JUNOS Software NETCONF API Guide*, GA32-0678
- ▶ *JUNOS Software Network Interfaces Configuration Guide*, GA32-0706
- ▶ *JUNOS Software Network Management Configuration Guide*, GA32-0698
- ▶ *JUNOS Software Policy Framework Configuration Guide*, GA32-0704
- ▶ *JUNOS Software Routing Protocols and Policies Command Reference*, GA32-0673
- ▶ *JUNOS Software Services Interfaces Configuration Guide*, GA32-0707
- ▶ *JUNOS Software Subscriber Access Configuration Guide*, GA32-0711
- ▶ *JUNOS Software System Basics and Services Command Reference*, GA32-0671
- ▶ *JUNOS Software System Log Messages Reference*, GA32-0675
- ▶ *JUNOS Software VPNs Configuration Guide*, GA32-0705
- ▶ *JUNOScope Software User Guide*, GA32-0670
- ▶ *JUNOS Software Policy Framework Configuration Guide*, GA32-0704

These publications from the Juniper Support website are also relevant as further information sources:

- ▶ For more information on IBM j-type s-series maintenance and analysis refer to *JUNOS Software Administration Guide*:
<http://www.juniper.net/techpubs/software/junos-security/junos-security10.1/junos-security-admin-guide/junos-security-admin-guide.pdf>
- ▶ For more information on Junos CLI commands refer to *JUNOS Software CLI Reference*:
<http://www.juniper.net/techpubs/software/junos-security/junos-security10.1/junos-security-cli-reference/junos-security-cli-reference.pdf>
- ▶ For more information on troubleshooting hardware components on IBM j-type s-series Ethernet Appliance J34S refer to *SRX3400 Services Gateway Hardware Guide*:
http://www.juniper.net/techpubs/en_US/release-independent/junos/information-products/topic-collections/hardware/srx-series/srx3400/book-srx3400-hw-ig.pdf

- ▶ For more information on troubleshooting hardware components on IBM j-type s-series Ethernet Appliance J36S refer to *SRX3600 Services Gateway Hardware Guide*:
http://www.juniper.net/techpubs/en_US/release-independent/junos/information-products/topic-collections/hardware/srx-series/srx3600/book-srx3600-hw-ig.pdf
- ▶ For more information on troubleshooting hardware components on IBM j-type s-series Ethernet Appliance J56S refer to *SRX5600 Services Gateway Hardware Guide*:
http://www.juniper.net/techpubs/en_US/release-independent/junos/information-products/topic-collections/hardware/srx-series/srx5600/book-SRX5600-hw-ig.pdf
- ▶ For more information on troubleshooting hardware components on IBM j-type s-series Ethernet Appliance J58S refer to *SRX5800 Services Gateway Hardware Guide*:
http://www.juniper.net/techpubs/en_US/release-independent/junos/information-products/topic-collections/hardware/srx-series/srx5800/book-srx5800-hw-ig.pdf

Online resources

These Web sites are also relevant as further information sources:

- ▶ IBM Juniper Partner website
<https://simplifymydatacenter.com/ibm>
- ▶ IBM j-type product page
<http://www.ibm.com/systems/networking/hardware/j-type/index.html>
- ▶ Juniper Networks Support site
<http://www.juniper.net/customers/support/>

How to get Redbooks

You can search for, view, or download Redbooks, Redpapers, Technotes, draft publications and Additional materials, as well as order hardcopy Redbooks publications, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

Numerics

1024-bit RSA private key 122
3DES (168-bit) 48, 61
4274-S34 43
4274-S36 43
4274-S56 55
4274-S58 55

A

AAA 276
AC power supply 77
access control list (ACL) 189
Accounting 207
Action modifiers 189
Active Alarms 396
Active configuration 103
active configuration 139
Active/active chassis cluster 293
Active/passive chassis cluster 278
add a new license key 426
address book 200
Address books 197
Address mapping 243
address range 225
address ranges 233
Address Resolution Protocol (ARP) 34, 276
address set names 206
address sets 200
Admin Status 344
Advanced Encryption Standard (AES) 247
AES encryption 48, 61
aggressive mode 248
AH 245, 250
air filter 87, 91
Alarm Index 341
Alarm indication signal (AIS) 390, 392
ALARM LED 388
Alarm LEDs 436
alarm relay 436
Alarm Terms 388
ALG 210
ambient temperature 90
application entry database 211
Application Layer Gateway (ALG) 210
application name 206
application names 206
application set 209
application set names 206
application timeout 211
application type 367
Application-Specific Integrated Circuits (ASICs) 103
Archiving 386
ARP 220
ASCII File 105

Assured forwarding (AF) 161
auditing elements 207
authentication certificate 121
Authentication Header (AH) 246, 249–250
Authentication method 374
Authorization 332
AutoKey IKE 244–245
Autokey IKE 251
Automated certificate enrollment (SCEP) 50, 62
Autonegotiation 14
autonomous system (AS) 342
Available capacity 149

B

backup configuration 401
Behavior aggregate 151
Behavior aggregate (BA) classifiers 153, 157
Best Effort 171
Best effort (BE) 161
BGP instances 50, 63
BGP peers 50, 63
BGP routes 50, 63
Border Gateway Protocol (BGP) 37
BPDU protection 29
Broadcast 20
Brute-force attack mitigation 48, 60

C

Cabling 72
Candidate configuration 103
candidate configuration 117, 139, 432
Categories 333
CBS 165
certificate authorities (CAs) 261
certificate authority (CA) 121
Chained Stateless Firewall Filters 189
Chassis and interfaces alarms 401
chassis and interfaces alarms 435
Chassis Cluster 275
chassis cluster 276
Chassis Cluster Interface 306
chassis cluster interface 292
Chassis Cluster Statistics 291, 307
Chassis Cluster Status 291–292, 306
Chassis Component Details 351
chassis components 403
Chassis Interface Monitoring 288, 299
Chassis Status 348
Chassis View 346
CIR 165
Class of Service (CoS) 149–150
Classless Inter-Domain Routing (CIDR) 35
class-of-service (CoS) 168
Cleanup 405

- clear-text password 113
- Clear-to-send (CTS) signal absent 391
- CLI 116
- client groups 267
- Coaxial Cabling 5
- Code Point Alias 157
- command completion 132
- command-line interface (CLI) 110
- commit command 146
- Committed Burst Size (CBS) 165
- Committed Information Rate (CIR) 165
- Common form-factor module (CFM) LED 435
- Community Name 332
- CompactFlash card 383, 422
- Compare Configuration 147
- Comparing configuration changes 432
- Concurrent VPN tunnels 48
- configuration files 401
- Configuration history 104
- Configuration Mode 106
- Configuration mode 130
- configuration mode 139
- configure exclusive 140
- configure private 140
- Configuring 122
- Configuring and Monitoring Alarms 388
- console access 116, 130
- Console port 84
- Contact Information 332
- Content integrity 244
- Context-Sensitive Help 137
- contiguous block 225
- contiguous block of addresses 232
- control link 276
- Control Plane 276
- control plane logs 381
- Control Port Configuration 296
- Cooling System 72
- cooling systems 87
- Copying files 405
- CoS 150, 161
- CPU Load 354
- Craft Interface LEDs 436
- Crossover Cable 7
- custom application 210
- Custom Application Mappings 210
- custom service 210

D

- Dashboard 125, 346
- Data 283
- Data carrier detect (DCD) signal absent 391
- Data Encryption Standard (DES) 247
- Data Fabric Configuration 296
- Data Link Layer 16
- data plane 277
- data plane logs 382
- Data set ready (DSR) signal absent 391
- data terminal equipment (DTE) 84
- Date From 387

- DB9 84
- DDoS 48, 60
- Default Classification 154
- default factory configuration 406
- default factory settings 402
- Default Gateway 38
- Default Route 38
- default router 111
- default timeout 211
- Delay jitter 149
- delete a license key 426
- Deleting files 405
- denial of service (DoS) 43
- denial-of-service (DoS) 116, 189, 249, 272
- deny 206
- DES (56-bit) 48, 61
- Destination address 220, 225
- destination address 206
- destination address names 206
- destination IP address 225
- Destination NAT 219, 225
- destination NAT address pool 225
- destination NAT address pools 225
- Destination NAT rules 220, 226
- Destination port 220
- destination port 367
- Destination Zone 374
- destination zone 206–207
- DH groups 246
- DHCP 51, 63, 276
- DHCP relay 51, 63
- Diagnosing tools 401
- Differentiated Services code point (DSCP) 153
- Diffie-Hellman (DH) 249
- Diffie-Hellman (DH) exchange 246
- Diffie-Hellman (DH) key 245
- directed broadcast messages 116
- Disabling System Logs 386
- Displaying installed license keys 427
- Displaying installed licenses 427
- Displaying license usage 427
- Distributed VPNs 249
- Distribution 104
- DNS server 111
- DOI 244
- Domain name 110
- Domain Name System (DNS) 200
- Domain of Interpretation (DOI) 244
- DoS 48, 60
- DOS attacks 163
- Downgrading Junos Software 424
- Drop probability 149
- Drop profiles 150–151
- DS1 (T1) 390
- DSCP for IPv6 DiffServ 153
- DSCPs 157
- dual Routing Engines 114
- Duplex 13
- Dynamic routing 51, 63
- Dynamic Services Architecture 43

Dynamic Virtual private network (VPN) 261
Dynamic VPN 261

E

EBS 165
egress interface 232
electrical specifications 73
Electromagnetic compatibility (EMC) certifications 65
electromagnetic interference (EMI) 82
electrostatic discharge (ESD) 74
Emacs 133
Enable Health Monitoring 334
Encapsulating Security Payload (ESP) 246–247, 249–250
Encryption algorithms 247
End-to-end packet delay 149
Equal-cost multipath (ECMP) 51, 63
Error Counters chart 344
ESP 245, 247, 250
event 378
Event ID 387
event ID 378
Events 377
Excess Burst Size (EBS) 165
Excessive number of zeros 393
Expedited forwarding (EF) 161
Extensible Markup Language (XML) 105

F

facility 378
Falling Threshold 334
Fan LEDs 436
fan tray 87, 89–90, 93
Far-end receive failure (FERF) 393
Fiber Optic Cabling 7
File system overview 401
File Usage 348
Filter-based forwarding (FBF) 51
Filtering Command Output 341
firewall 38
firewall authentication history 375
firewall filter 189, 191
Firewall packets per second 47, 59
Firewall performance 47
Flexible PIC concentrators 196
Forwarding Classes 150
Forwarding classes 151
forwarding classes 161
Frame Check Sequence (FCS) 16
Frame Forwarding 21
Frame Structure 16
FTP 110, 116, 262, 409
Functional Zones 196

G

gates 373
Generating a license key 425
Gigabit Interface Converter (GBIC) 11

Graphical Chassis 125
Grounding 74

H

Halt 402
halting 401
Hardware or software failure 390
Hash Message Authentication Code (HMAC) 246
Help Reference 138
Help Topic 138
hierarchical configuration 140
Host subsystem LEDs 436
hostname 110
HTTP 52, 120, 122, 262
HTTP Access 123
HTTP on All Interfaces 123
HTTP over Secure Sockets Layer (HTTPS) 105
HTTP Web Access 123
HTTPS 52, 120, 122
HTTPS Access 123
HTTPS Certificate 123
HTTPS on All Interfaces 123
HTTPS Web Access 123
Hub 11
HyperTerminal 111
Hypertext Transfer Protocol (HTTP) 105

I

I/O Cards (IOCs) 45
I/O cards (IOCs) 43
IBM Ethernet Appliance J34S 41, 44
IBM Ethernet Appliance J36S 41, 44
IBM Ethernet Appliance J56S 41, 55
IBM Ethernet Appliance J58S 41, 55
IBM j-type Ethernet appliance 41
ICMP 189, 205, 211, 311
ICMP Address Sweep 361
ICMP Flood 361
ICMP Fragment 362
ICMP Large Packet 362
ICMP Ping of Death 361
Idle alarm 393
IDP Memory Statistics 364
IEEE 802.1ad drop eligible indicator (DEI) bit 153
IEEE 802.1p CoS bits 153
IETF 153
IKE 250–251
IKE gateway 251, 257
IKE IPsec tunnel negotiation 251
IKE negotiations 244
IKE policy 251, 257
IKE proposal 257
IKE Tunnel Negotiation 248
Inactivity timeout 243
Inbound Traffic 198
incoming zone 207
Initialization Process 102
Input Rate graph 344
Integrated Routing and Bridging (IRB) 289

- Intelligent Drop Mechanisms (WRED) 51, 64
- Interface Ports 196
- Interface Process 102
- Interface source NAT 62
- Interfaces 197
- interframe gap (IFG) 17
- Internet Control Message Protocol (ICMP) 342
- Internet Engineering Task Force (IETF) 36
- Internet Engineering Task Force (IETF) 328
- Internet Group Management Protocol (IGMP) 23
- Internet Protocol Security (IPsec) 244
- Internet Service Provider (ISP) 3
- Interval 334
- Intrusion Detection and Prevention (IDP) 271, 363
- IOC and SPC LEDs 436
- IOC LED 437
- IOCs 57
- IP Address Classes 35
- IP address shifting 233
- IP Bad Options 362
- IP Block Fragment 363
- IP Loose route Option 362
- IP precedence bits 153
- IP Record Route Option 362
- IP Routing 33
- IP Security Option 362
- IP Source Route 361
- IP Spoofing 361
- IP Stream Option 362
- IP Strict Source Route Option 362
- IP Tear Drop 361
- IP Timestamp Option 362
- IP Unknown Protocol 362
- IPsec 244
- IPSec policy 258
- IPSec proposal 258
- IPsec security associations (SAs) 277
- IPsec tunnel 247
- IPsec VPN 48
- IPsec VPN tunnel policies 206
- IPv4 addresses 34
- IPv6 Addressing 36
- ISAKMP 250

J

- J34S 42
- J36S 42
- J56S 55
- J58S 55
- Jumbo frame 17
- Junos 66, 99
- Junos CLI 106, 130
- Junos command line interface (CLI) 119
- Junos Command-Line Interface (CLI) 105
- Junos Operating System 41, 66
- Junos Software 100, 420
- Junos Software File System 428
- Junos Software stateful firewall policy 205
- JUNOScope 124
- JUNOScript API 105

- JUNOScript client 124
- JUNOScript over SSL 122
- JUNOScript XML scripting API 124
- junos-global zone 197
- J-Web 105, 120
- J-Web Event Viewer 386
- J-Web graphical user interface (GUI) 105, 117, 119
- J-Web Quick Configuration 328, 331
- J-Web user interface 331

L

- L3 sub interfaces 50
- Layer 1 4
- Layer 2 16
- Layer 3 32
- LC connector 9
- LDAP authentication 265
- LED 73, 80
- LED indicator 77
- License data 425
- License ID 425
- license key 425
- Licenses 128
- Line Cards 57
- line cards 44
- Line code violation 393
- Link aggregation 27
- Link Layer Discovery Protocol (LLDP) 30
- Link Layer Discovery Protocol–Media Endpoint Discovery (LLDP-MED) 30
- Link Status 344
- LLDP 30
- LLDP-MED 30
- Load Configuration 147
- Loading a configuration from a file 434
- Local Area Network (LAN) 3
- Local authentication 265
- Local Engine ID 332
- Login Sessions 348
- Loop protection 29
- Loss of frame (LOF) 393
- Loss of receive clock 391
- Loss of signal 393
- Loss of signal (LOS) 392
- Loss of transmit clock 391
- loss priority 151

M

- MAC address 17
- main mode 248
- Maintain 128
- Malformed packet protection 48, 60
- Manage files 402
- manage Junos Software licenses 425
- Management (MGMT) 85
- management (MGT) zone 196
- Management Information Base 328
- Management Information Base (MIB) 328
- management interface 110

- Management Process 102
- managing files 401
- Managing licenses 401
- Manual key 245
- match criteria 206
- Maximum 3DES+SHA-1 VPN performance 47, 59
- Maximum AES256+SHA-1 VPN performance 47, 59
- Maximum available slots for IOCs 60
- Maximum available slots for NPCs 48
- Maximum available slots for SPCs 47, 60
- Maximum concurrent sessions 47, 59
- Maximum Firewall Performance 59
- Maximum IPS performance 47, 59
- Maximum number of security zones 63
- Maximum number of virtual routers 63
- Maximum number of VLANs per interface 63
- Maximum security policies 47, 59
- Maximum session number 243
- Maximum user supported 47, 59
- MD5 48, 61
- Media Access Control(MAC) 16
- Medium Dependent Interface Crossover(MDIX) 7
- Medium Dependent Interface(MDI) 7
- Memory Utilization 354
- Message Digest 5 (MD5) 246
- MIBs 328, 335, 338
- Monitor 127
- Monitor interface 414
- Monitor traffic 416
- monitoring alarms 345
- Monitoring Chassis Information 345, 349
- Monitoring Firewall Authentication 373
- Monitoring Flow Session Statistics 364
- Monitoring IDP 363
- monitoring interface 401
- Monitoring Interfaces 343
- monitoring policies 356
- Monitoring Process Details 345, 353
- Monitoring Screen Counters 360
- Monitoring system commands 418
- Monitoring System Properties 345
- most recently committed configuration 431
- MPLS EXP bits 153
- MPLS routing table 105
- Multicast 19
- Multicast routing table 105
- Multifield (MF) classifiers 153
- Multifield Classifier (MF) 159
- Multifield traffic classifiers 151
- Multi-mode fiber (MMF) 8
- Multiple Spanning Tree Protocol (MSTP) 27
- Multiple VLANs Mode 26

N

- NAT 49, 61, 218, 242
- NAT binding 243
- NAT proxy ARP 220
- NAT Rule Sets 219
- NAT rules 220
- NETCONF API 106

- Network Address Translation (NAT) 43
- Network attack detection 48, 60
- Network Control 171
- Network Control (NC) 161
- Network Firewalls 38
- Network Interface Card (NIC) 17
- network management systems (NMSs) 328
- Network Processing Cards (NPCs) 45
- network processing cards (NPCs) 43
- network security 187
- network utilities 401, 408
- New sessions/second 47, 59
- Node Specific Configuration 296

O

- object ID (OID) 328
- OID 328
- One Modular Software Architecture 67
- One modular software architecture 100
- One Operating System 67
- One operating system 100
- One Software Release 67
- One software release train 100
- Open Shortest Path First (OSPF) 37, 380
- Open System Interconnect (OSI) 2
- Operational Mode 106
- Operational mode 130–131
- Optical Ports 86
- Organizationally Unique Identifier(OUI) 18
- Original Equipment Manufacturer (OEM) 41
- OSPF 211
- OSPF Configuration 302
- OSPF instances 50, 63
- OSPF neighbor 308
- OSPF protocol 302
- OSPF routes 50, 63
- Out of frame (OOF) 392
- outgoing zone 207
- Output Rate graph 344
- Overflow pool 233
- Oversubscribed NAT pool 49, 62

P

- Packet classification 150–151
- Packet Counters chart 344
- Packet Forwarding Engine 101
- Packet Forwarding Engine (PFE) 103, 276
- Packet information 221, 226, 234
- Packet loss priority 150
- packet loss priority (PLP) 165
- Packet's Loss Priority (PLP) 170
- Pass-Through Authentication 262
- pass-through user authentication 262
- password recovery 407
- PAT 49, 61, 233
- Peak Information Rate (PIR) 165
- Per port limit (PPL) 14
- Perfect forward secrecy 48, 61
- Perfect Forward Secrecy (PFS) 249

- permit 206
- Persistent NAT 241
- persistent NAT 243
- persistent NAT bindings 242
- Personal firewalls 38
- Phase-locked loop out of lock 393
- Physical Interface Cards(PIC) 196
- PIC Name 364
- PID 354
- PING 310, 323
- ping 408
- pinholes 373
- Pipe Commands 134
- PIR 165
- PKI 50, 62
- PLP 170
- PoE 85
- Policers 151
- Policies 197
- Policing 150, 165
- Policy Id 374
- Policy list 356
- policy name 206
- Policy-based routing 63
- Port Address Translation (PAT) 233
- port address translation (PAT) 221
- port forwarding 225
- Port module LED 437
- Port Randomization 237
- port translation 233
- Port VLAN ID (PVID) 26
- Power 72
- Power and Fan Tray Details 351
- Power over Ethernet (PoE) 14, 85
- Power supply LEDs 436–437
- Preliminary zone 257
- Preshared Key 247
- Prevent replay attack 48, 61
- previous configurations 431
- Priority 378
- Priority Scheduling 172
- Privacy 244
- privacy-enhanced mail (PEM) 122
- process ID 379
- Protocol mode 244
- protocol-based default timeout 211
- Proxy ARP 49, 62
- Public key infrastructure (PKI) 261
- public-private key 121
- putty.exe 111

Q

- QoS 55
- quality of Service (QoS) 43
- query binding 243

R

- Racks 72
- RADIUS accounting 50, 62

- RADIUS authentication 265
- Random Early Detection (RED) 175
- random early detection (RED) 150
- Rapid Spanning Tree Protocol (RSTP) 27
- RE LEDs 437
- Reboot 128, 402
- rebooting 401
- Reconnaissance Deterrence 271
- recovery software package 421
- RED 161, 175
- RED Profile 175
- Redbooks Web site 441
 - Contact us xiv
- Redundancy group 285
- Redundancy Group Configuration 296
- Redundancy groups 277
- Redundant Ethernet (RETH) interface 286
- redundant Ethernet interface 278
- Redundant System Log Server 380
- Redundant VPN gateways 48, 61
- reflexive transport addresses 243
- reject 206
- Remote access VPN 48, 61
- Remote defect indication 392
- remote host 241
- Remote management access 116
- Repeater 12
- Replay Protection 249
- rescue configuration 402, 406
- RESET flag 199
- resource manager 369
- Resource Utilization 125, 347
- Restart software 402
- RETH 286, 298
- Reverse mapping 220
- Reverse path forwarding (RPF) 51, 63
- Rewrite Rules 168
- Rewrite rules 150–151
- RIP 50, 63
- Rising Threshold 334
- RJ45 84
- RMON 329
- Rollback Command 146
- root password 402
- Root protection 29
- Route lookup 220
- Route Table 37
- Routing Engine 101, 189
- Routing Engine Kernel 102
- Routing Engine LEDs 435
- Routing Information Protocol (RIP) 37
- Routing instance 225, 233
- routing instance 315
- Routing Protocol Process 102
- Routing Protocols 37
- Routing Table 322
- routing-instance 226
- routing-instances 318
- RS-232 84
- RSA SecurID 265

- RSA/DSA certificates 247
- RSTP 28
- RTOs 277
- RTP/RTCP 234
- Run Command 147
- runtime objects (RTOs) 277

S

- SA 251
- Save Configuration 146
- Saving a configuration to a file 433
- Saving licenses keys 426
- Saving the configuration 430
- SC connector 10
- SCB LEDs 436–437
- Scheduler 151
- Scheduler Buffer Size 171
- Scheduler drop profile 172
- Schedulers 150
- schedulers 171
- Screens 197
- Secure Hash Algorithm (SHA-1) 246
- secure management 121
- Secure Shell (SSH) 130
- Secure Sockets Layer (SSL) 121
- Secure Web Access 121
- SecurID authentication 265
- Security algorithms 244
- security associations (SAs) 244
- security domains 211
- Security parameter index (SPI) 245
- security parameter index (SPI) 244
- security policies 205
- security policy 320
- Security Policy Applications 209
- Security Policy Configuration 301
- Security policy lookup 220
- security policy lookup 220
- Security Policy Schedulers 208
- Security Processing Units (SPUs) 250
- Security Resources 125, 348
- security zone 205, 315, 320
- security zones 50, 195, 318
- Sender authentication 244
- Service Processing Cards (SPCs) 44
- service processing cards (SPCs) 43
- service-level agreement (SLA) 163
- Services 344
- Services Processing Card (SPC) 43, 282
- Session Initiation Protocol (SIP) 242
- Session Traversal Utilities for NAT (STUN) 241–242
- severity level 379, 381
- SFB LEDs 435
- SHA-1 48, 61
- Shaping rate 172
- Shared Media Segment 12
- Shielded Twisted-Pair (STP) 5
- show commands 411
- simple filter 163
- Simple Network Management Protocol (SNMP) 328
- single IP address 225
- Single mode fiber (SMF) 8
- Single VLAN Mode 26
- Single-rate tricolor marking 165
- site-to-site VPN 244
- SNMP 52, 65, 116, 118, 328
- SNMP Agents and Communities 335
- SNMP communities 329, 337
- SNMP configuration 339
- SNMP Health Monitor 329
- SNMP health monitor 340
- SNMP Process 103
- SNMP Quick Configuration 331
- SNMP Trap Groups 335
- SNMP traps 329
- software license keys 425
- software package 421
- software process 404
- source address 206
- source address names 206
- Source interface 219
- Source NAT 219, 232
- source NAT address pool 233
- Source NAT Pools 233
- Source NAT rules 220
- source Network Address Translation (NAT) 355
- Source pool grouping 49, 61
- Source pool utilization alarm 49, 61
- source port 371
- source port range 209
- Source routing instance 219
- Source Zone 374
- Source zone 219
- source zone 206–207
- Spanning Tree Protocol (STP) 27
- SPC 55, 276
- SPC LED 437
- Spoofing 329
- SPUs 250
- SRX Clustering Module (SCM) LED 436
- SRX3400 43
- SRX3600 43
- SRX5600 55
- SRX5800 55
- SSH 52, 110, 116, 409
- SSH public key 113
- SSL 121
- SSL certificate 122
- SSL certificates 122
- SSL encryption 122
- SSL server certificate 121–122
- Start Frame Delimiter (SFD) 16
- stateful firewall 205
- Statefull Firewall 38
- Stateless Firewall 39
- stateless firewall filters 188
- Static NAT 219, 221
- Static NAT rules 220
- Static routes 51, 63
- Static Source NAT 49, 61

- Storage Usage 348
- Straight 6
- Structure of Management Information (SMI) 328
- STUN 241
- Sub-Component 353
- Subnet Mask 36
- Switched Segment 13
- Symmetric NAT 49, 62
- SYN cookie protection 48, 60
- SYN flood attack 271
- Synchronization 104
- SYNchronize flag 199
- syslog messages 377
- System Alarms 348
- System Description 332
- System Identification 125, 347
- System Identification Information 335
- System Location 332
- System Log Facilities 381
- system log messages 377
- system logs 420
- System Name Override 332
- System name override 335
- system snapshot 421

T

- T3 (DS3) 393
- TACACS 266
- Target host 241
- Target host port 242
- Targets 333
- TCP 211
- TCP FIN without ACK 362
- TCP Land Attack 361
- TCP No Flag 362
- TCP Port Scan 361
- TCP reassembly 48, 60
- TCP RST 205
- TCP segment headers 271
- TCP SYN 271
- TCP SYN Attack 361
- TCP SYN FIN Packet 362
- TCP SYN Fragment 362
- TCP SYN-ACK-ACK Proxy 363
- TCP Winnuke 361
- TCP-Reset Parameters 199
- TCP-RST 197, 199
- Telnet 110, 116, 130, 262, 409
- telnet 52
- Temperature 352
- terminal session 384
- TIA-598C standard 9
- timeout value 210
- Total IDP Data Plane Memory (MB) 364
- trace files 420
- Traceroute 52, 65
- traceroute 313, 408
- Traffic direction 221, 226, 234
- transceiver 10
- Transmission Rate 171

- Transparent mode 273
- transport protocol 210
- Trap Group Name 332
- Triple DES (3DES) 247
- Trust zone 197
- Tunnel interfaces 60
- tunnel mode 250
- tunnel session 372
- Twisted-Pair Cabling 5
- Twisted-pair crossover 7
- Two-color marking 165
- two-rate tricolor marker (TCM) 163
- Two-rate tricolor marking 165
- Type Length Value (TLV) 30

U

- UAC enforcement 62
- UAC enforcement point 50
- UDP 211, 250
- UDP Flood 361
- UDP session 271
- Unicast 19
- Unicast Frame 21–22
- Unicast routing table 104
- Unshielded Twisted-Pair (UTP) 5
- update a license key 426
- Upgrading Junos Software 422
- User Datagram Protocol (UDP) 242

V

- Virtual Local Area Network (VLAN) 23
- Virtual Private Network (VPN) 43
- virtual private network (VPN) 244
- Virtual Router 275
- virtual routers 50, 315
- virtual systems (vsys) 211
- VLAN Spanning Tree Protocol (VSTP) 27
- VLANs per interface 50
- VPN tunnel 258
- VSTP 28

W

- Web Authentication 262
- Web authentication 263
- Web-based authentication 50, 62
- WELF log collector 382
- WELF logs 383
- WELF-formatted logs 382
- Wide Area Network (WAN) 3

Z

- Z-mode deployment 293
- Zone 344
- Zone Configuration 288, 300
- Zone-based IP spoofing 48, 60



Implementation of IBM j-type Ethernet Appliances

(1.0" spine)

0.875" <-> 1.498"

460 <-> 788 pages



Redbooks®

Implementation of IBM j-type Ethernet Appliances

Introduction to IBM j-type hardware

Introduction to Junos and CLI

Instructions for planning, installing, and configuring hardware

IBM j-type data center solutions that run Junos software provide operational agility and efficiency that dramatically simplifies the network and delivers unprecedented savings. This solution enables a network design with fewer devices, interconnections, and network tiers. Beyond the obvious cost advantages, the design enables the following key benefits:

- ▶ Reduces latency
- ▶ Simplifies device management
- ▶ Delivers significant power, cooling, and space savings
- ▶ Eliminates multiple system failure points
- ▶ Performs pervasive security

The high-performance data center is built around IBM j-type e-series switches, m-series routers, and s-series firewalls. It is a new family of powerful products that help shape the next-generation dynamic infrastructure.

IBM j-type s-series Ethernet Appliances perform essential networking security functions and are ready for next-generation data center services and applications. Designed on top of the Junos operating system, s-series Ethernet Appliances provide flexible processing scalability, I/O scalability, network segmentation, and services integration.

In this Redbooks publication, we target IT professionals who sell, design, or administer IBM j-type networking solutions and cover the basic installation and maintenance of the IBM j-type s-series Ethernet Appliance hardware.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-7883-00

ISBN 0738435031