

Operating Juniper Networks Routers in the Enterprise

8.a

Student Guide



1194 North Mathilda Avenue
Sunnyvale, CA 94089
USA
408-745-2000
www.juniper.net

Course Number: EDU-JUN-OJRE

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

Operating Juniper Networks Routers in the Enterprise Student Guide, Revision 8.a

Copyright © 2007, Juniper Networks, Inc.

All rights reserved. Printed in USA.

Revision History:

Revision 8.a—March 2007

The information in this document is current as of the date listed above.

The information in this document has been carefully verified and is believed to be accurate for software Release 8.1R2. Juniper Networks assumes no responsibilities for any inaccuracies that may appear in this document. In no event will Juniper Networks be liable for direct, indirect, special, exemplary, incidental or consequential damages resulting from any defect or omission in this document, even if advised of the possibility of such damages.

Juniper Networks reserves the right to change, modify, transfer or otherwise revise this publication without notice.

YEAR 2000 NOTICE

Juniper Networks hardware and software products do not suffer from Year 2000 problems and hence are Year 2000 compliant. The JUNOS software has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

SOFTWARE LICENSE

The terms and conditions for using this software are described in the software license contained in the acknowledgment to your purchase order or, to the extent applicable, to any reseller agreement or end-user purchase agreement executed between you and Juniper Networks. By using this software, you indicate that you understand and agree to be bound by those terms and conditions.

Generally speaking, the software license restricts the manner in which you are permitted to use the software and may contain prohibitions against certain uses. The software license may state conditions under which the license is automatically terminated. You should consult the license for further details.

Contents

Chapter 1:	Course Introduction	1-1
Chapter 2:	Juniper Networks Enterprise Routers	2-1
	Customer Edge and Enterprise Platforms	2-3
	Overview of Enterprise Routing Platforms	2-8
	Architecture and Packet Flow	2-15
	Interface Support and Naming	2-37
	Field-Replaceable Units	2-43
	Network Management Options	2-47
Chapter 3:	JUNOS User Interfaces	3-1
	User Interface Options	3-3
	User Authentication and Authorization	3-6
	Active and Candidate Configurations	3-15
	Using the J-Web Graphical User Interface	3-18
	Lab 1, Parts 1–3: The J-Web Interface	3-35
	Using the JUNOS Software Command-Line Interface	3-36
	Lab 1, Parts 4–5: The JUNOS Software CLI	3-78
Chapter 4:	Installation and Initial Configuration	4-1
	Installation Guidelines	4-3
	Autoinstallation	4-8
	Rescue and Factory-Default Configurations	4-19
	Configuration Checklist	4-26
	Initial Configuration Using J-Web	4-29
	Initial Configuration Using the CLI	4-38
	Overview of Interface Configuration	4-49
	Configuring Interfaces Using J-Web	4-56
	Lab 2: Initial Configuration	4-73
Chapter 5:	Operational Monitoring and Maintenance	5-1
	Monitoring Platform Operation	5-3
	Monitoring Interface Operation	5-9
	Network Utilities	5-25
	System Logging and Protocol Tracing	5-31
	License Management	5-42
	Maintaining JUNOS Software	5-48
	File System Maintenance and Password Recovery	5-63
	Lab 3: Operational Monitoring	5-71

Chapter 6:	Routing Protocols and Policy	6-1
	Routing Tables and Route Preferences	6-3
	Routing Policy	6-7
	J-Web Support for Routing Protocols and Policy	6-16
	Configuring and Monitoring Static Routing	6-21
	Interior Gateway Protocols	6-28
	Configuring and Monitoring RIP	6-31
	Lab 4, Parts 1–3: RIP	6-47
	Configuring and Monitoring OSPF	6-48
	Lab 4, Parts 4–5: OSPF	6-72
	Configuring and Monitoring Basic BGP	6-73
	Lab 5: Static and BGP Routing	6-93
 Chapter 7:	 Adaptive Services	 7-1
	Overview of Adaptive Services Features and Architecture	7-3
	Configuration and Monitoring of Packet Filters	7-8
	Configuration and Monitoring of Stateful Firewalls	7-12
	Configuration and Monitoring of NAT/PAT	7-26
	Configuration and Monitoring of IPSec Tunnels	7-37
	Overview of Intrusion Detection System Capabilities	7-44
	Overview of Flow Monitoring and Accounting	7-46
	Overview of J-series CoS Support	7-48
	Lab 6: Services	7-58
 Appendix A:	 Supported PIMs	 A-1
 Appendix B:	 New Features	 B-1

Course Overview

This three-day course is an introductory-level, instructor-led course that focuses on installation, configuration, and operational analysis of Juniper Networks routers in the enterprise environment. OJRE introduces Juniper Networks enterprise routing platforms including both M-series and J-series models. It then focuses on router configuration using both the J-Web graphical user interface (GUI) and the JUNOS software command-line interface (CLI). Real-world configuration and operational monitoring case studies are provided for general router configuration and for RIP, static, and OSPF routing. The class also provides an overview of common services such as IPSec VPNs and stateful firewall/NAT.

The course combines both lecture and labs, with significant time allocated for hands-on experience with J-series platforms and JUNOS Internet software. The OJRE class is an excellent way to prepare students for attending other offerings in the Juniper Networks training curriculum.

Objectives

After successfully completing this course, you should be able to install, configure, and operate J-series platforms.

Intended Audience

The primary audiences for this course are end users of J-series platforms, which include the following:

- Network engineers;
- Support personnel;
- Reseller support; and
- Others responsible for implementing Juniper enterprise routing products.

Course Level

OJRE is an introductory-level course.

Prerequisites

The OJRE prerequisite is a basic understanding of the TCP/IP protocols.

While not required, familiarity with the command-line interface of a routing platform or UNIX system is helpful.

Course Agenda

Day 1

- Chapter 1: Course Introduction
- Chapter 2: Juniper Networks Enterprise Routers
- Chapter 3: JUNOS User Interfaces
- Chapter 4: Installation and Initial Configuration

Day 2

- Chapter 5: Operational Monitoring and Maintenance
- Chapter 6: Routing Protocols and Policy

Day 3

- Chapter 7: Adaptive Services

Document Conventions

CLI and GUI Text

Frequently throughout this course, we refer to text that appears in a command-line interface (CLI) or a graphical user interface (GUI). To make the language of these documents easier to read, we distinguish GUI and CLI text from chapter text according to the following table.

Style	Description	Usage Example
Franklin Gothic	Normal text.	Most of what you read in the Lab Guide and Student Guide.
Courier New	Console text: <ul style="list-style-type: none">Screen capturesNoncommand-related syntax	<code>commit complete</code> <code>Exiting configuration mode</code>
Century Gothic	GUI text elements: <ul style="list-style-type: none">Menu namesText field entry	Select File > Open, and then click Configuration.conf in the Filename text box.

Input Text Versus Output Text

You will also frequently see cases where you must enter input text yourself. Often this will be shown in the context of where you must enter it. We use bold style to distinguish text that is input versus text that is simply displayed.

Style	Description	Usage Example
Normal CLI	No distinguishing variant.	Physical interface:fxp0, Enabled
Normal GUI		View configuration history by clicking Configuration > History.
CLI Input GUI Input	Text that you must enter.	lab@San_Jose> show route Select File > Save, and enter config.ini in the Filename field.

Defined and Undefined Syntax Variables

Finally, this course distinguishes between regular text and syntax variables, and it also distinguishes between syntax variables where the value is already assigned (defined variables)

and syntax variables where you must assign the value (undefined variables). Note that these styles can be combined with the input style as well.

Style	Description	Usage Example
<i>CLI Variable</i>	Text where variable value is already assigned.	<code>policy my-peers</code>
<i>GUI Variable</i>		Click on <i>my-peers</i> in the dialog.
<u><i>CLI Undefined</i></u>	Text where the variable's value is the user's discretion and text where the variable's value as shown in the lab guide might differ from the value the use must input.	Type set policy <u>policy-name</u> .
<u><i>GUI Undefined</i></u>		Select File > Save, and enter <u>filename</u> in the Filename field.

Additional Information

Education Services Offerings

You can obtain information on the latest Education Services offerings, course dates, and class locations from the World Wide Web by pointing your Web browser to:
<http://www.juniper.net/training/education/>.

About This Publication

The *Operating Juniper Networks Routers in the Enterprise* Student Guide was developed and tested using software version 8.1R2. Previous and later versions of software may behave differently so you should always consult the documentation and release notes for the version of code you are running before reporting errors.

This document is written and maintained by the Juniper Networks Education Services development team. Please send questions and suggestions for improvement to training@juniper.net.

Technical Publications

You can print technical manuals and release notes directly from the Internet in a variety of formats:

- Go to <http://www.juniper.net/techpubs/>.
- Locate the specific software or hardware release and title you need, and choose the format in which you want to view or print the document.

Documentation sets and CDs are available through your local Juniper Networks sales office or account representative.

Juniper Networks Support

For technical support, contact Juniper Networks at <http://www.juniper.net/customers/support/>, or at 1-888-314-JTAC (within the United States) or 408-745-2121 (from outside the United States).



Operating Juniper Networks Routers in the Enterprise

Chapter 1: Course Introduction

Chapter Objectives

- After successfully completing this chapter, you will be able to:
 - Get to know one another
 - Identify the objectives, prerequisites, facilities, and materials used during this course
 - Identify additional Juniper Networks courses
 - Describe the Juniper Networks Technical Certification Program (JNTCP)



This Chapter Discusses:

- Objectives and course content information;
- Additional Juniper Networks courses; and
- Juniper Networks Technical Certification Program.

Introductions

- What is your name?
- Where do you work?
- What is your primary role in your organization?
- What kind of network experience do you have?
- What is the most important thing for you to learn in this training session?



Introductions

This slide serves to break the ice by having you introduce yourself and state your reasons for attending the class.

Course Contents

- Chapter 1: Course Introduction
- Chapter 2: Juniper Networks Enterprise Routers
- Chapter 3: JUNOS User Interfaces
- Chapter 4: Installation and Initial Configuration
- Chapter 5: Operational Monitoring and Maintenance
- Chapter 6: Routing Protocols and Policy
- Chapter 7: Adaptive Services
- Appendix A: Supported PIMs
- Appendix B: New Features



Course Contents

This slide lists the topics for this course.

Prerequisites

- The prerequisites for this course are the following:
 - Basic understanding of the TCP/IP protocols
 - While not required, familiarity with the command-line interface of a routing platform or UNIX system is helpful

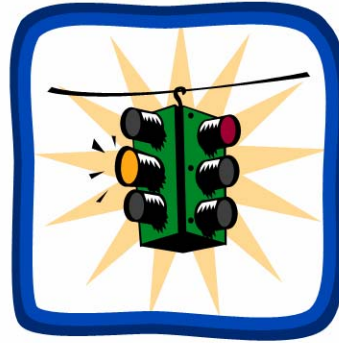


Prerequisites

This slide lists the prerequisites for this course.

Course Administration

- Course objectives
- Sign-in sheet
- Schedule
 - Class times
 - Breaks
 - Lunch
- Break and restroom facilities
- Communications
 - Telephones
 - Cellular phones and pagers
 - Internet access



General Course Administration

This slide documents general aspects of classroom administration.

Education Materials

- Available in class:

- Lecture material
- Lab guide
- Lab equipment

- Available outside of class:

- Online documentation at www.juniper.net
- Juniper Networks Technical Assistance Center (JTAC)

- Available through your account representative:

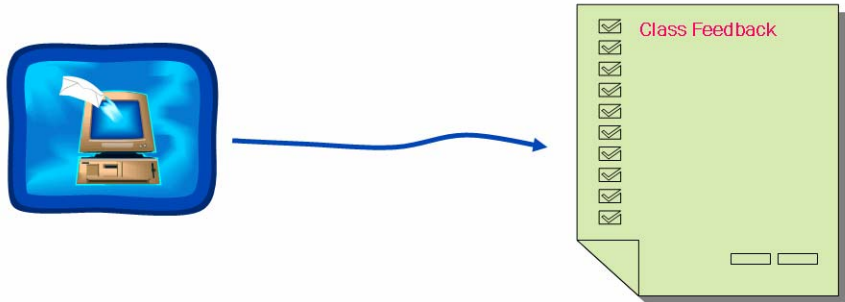
- Documentation CD
- Printed documentation



Training and Study Materials

This slide describes several options for obtaining study and preparation materials.

Satisfaction Feedback



- Please be sure to tell us how we did!
 - You will receive a survey to complete either at the end of class, or we will send it to you by e-mail within two weeks
- Completed surveys:
 - Help us serve you better
 - Ensure that you receive a certificate of completion

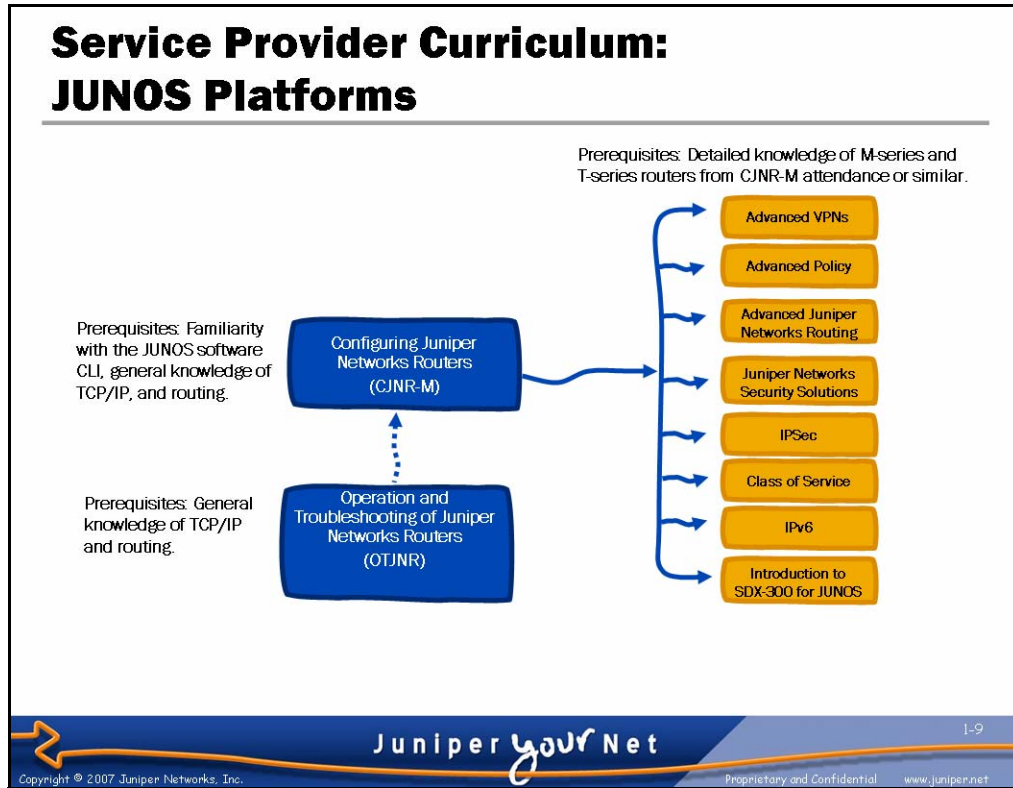
Juniper your Net

Copyright © 2007 Juniper Networks, Inc. Proprietary and Confidential www.juniper.net 1-8

Satisfaction Feedback

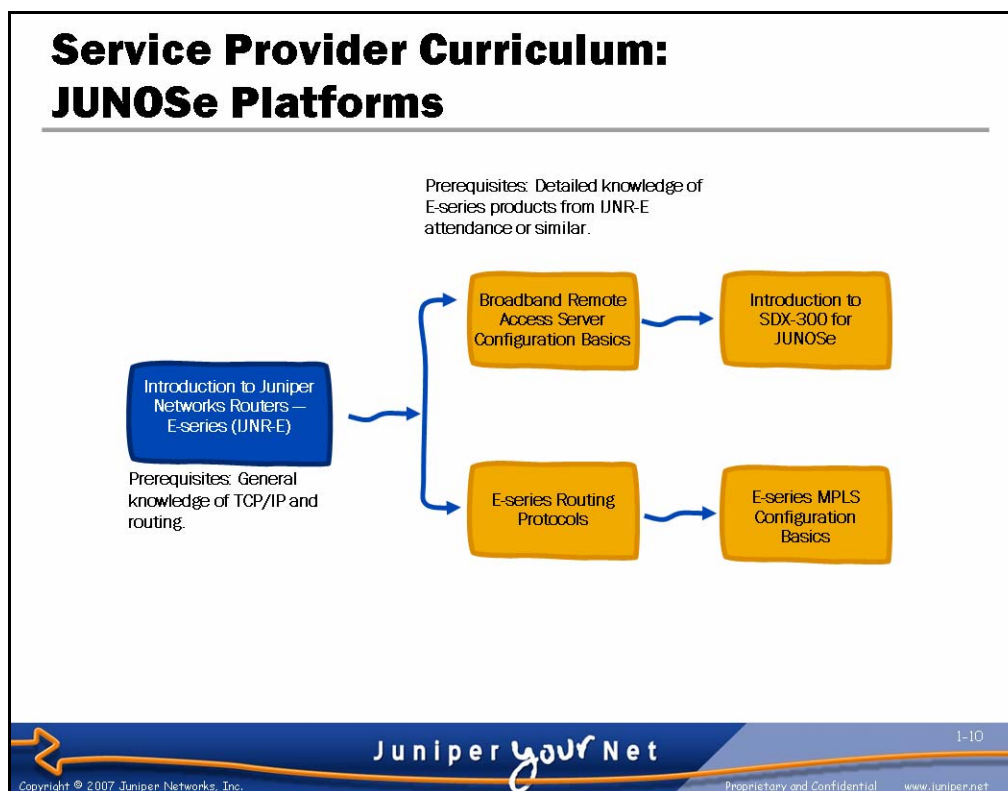
Juniper Networks uses an electronic survey system to collect and analyze your comments and feedback. Depending on the class you are taking, please complete the survey at the end of the class, or be sure to look for an e-mail about two weeks from class completion that directs you to complete an on-line survey form (be sure to provide us with your current e-mail address).

Submitting your feedback entitles you to a certificate of class completion. We thank you in advance for taking the time to help us improve our educational offerings.



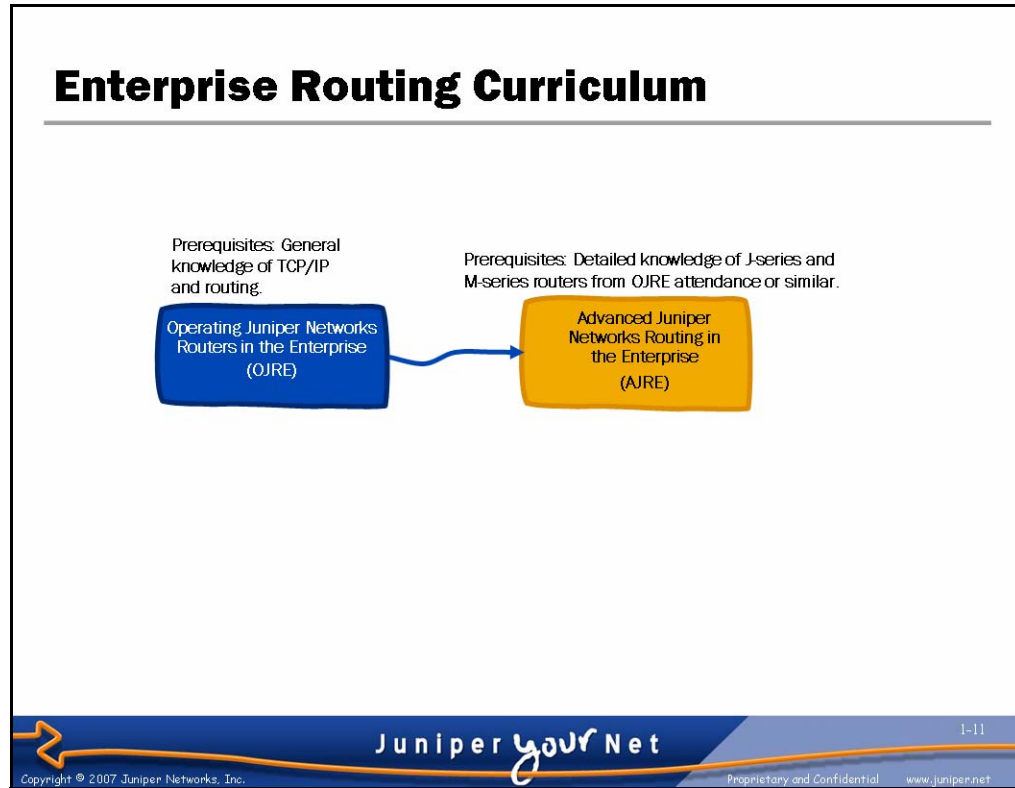
M-series, T-series, and J-series Curriculum

This graphic displays the primary Education Services offerings that support Juniper Networks M-series, T-series, and J-series technologies.



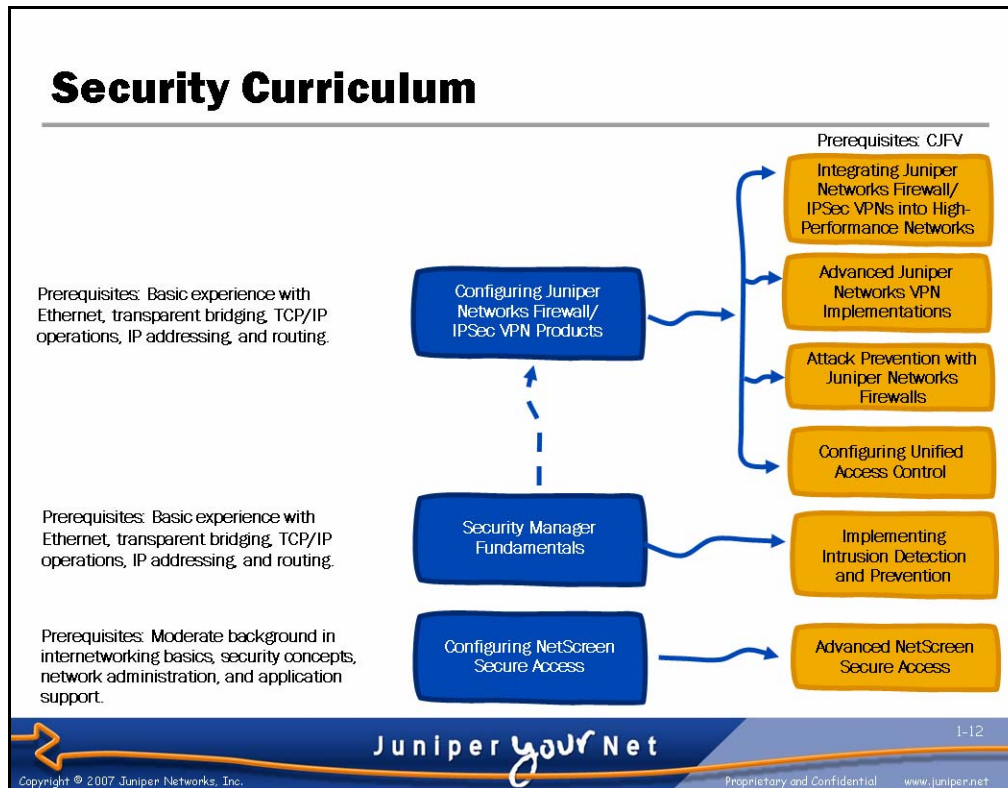
E-series Curriculum

This graphic displays the primary Education Services offerings that support Juniper Networks E-series router technologies.



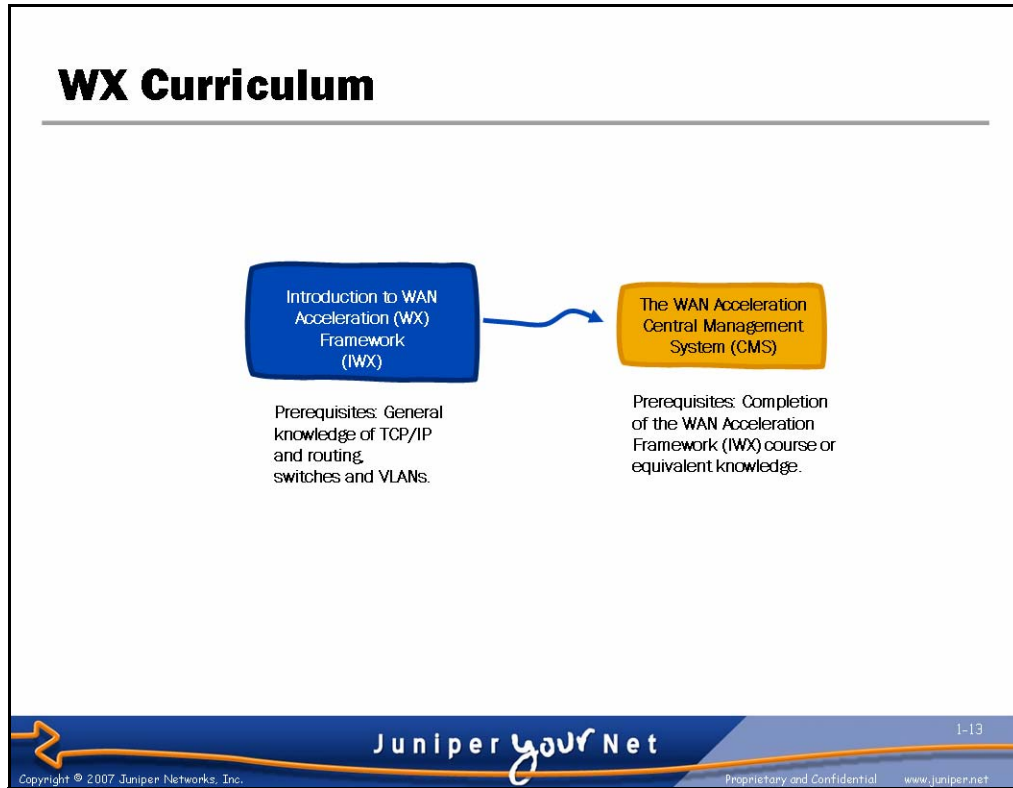
Security Curriculum

This graphic displays the primary Education Services offerings that support Juniper Networks security technologies.



WX Curriculum

This graphic displays the primary Education Services offerings that support Juniper Networks WX Framework technologies.



DX Curriculum

This graphic displays the primary Education Services offerings that support Juniper Networks DX Application Acceleration Platform technologies.

DX Curriculum

Implementing the DX
Application
Acceleration Platform
(IDX)

Prerequisites: General
knowledge of TCP/IP,
HTTP, and SSL.

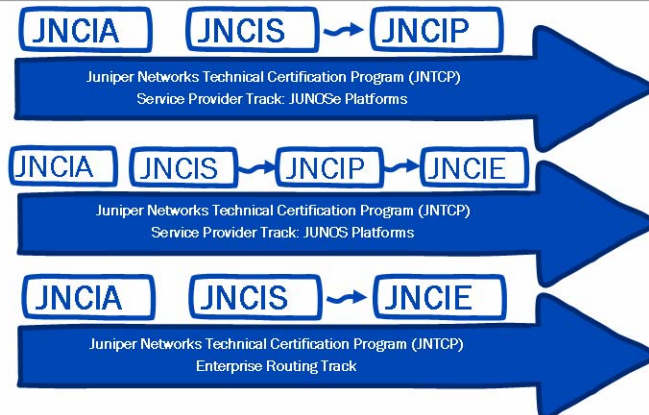
1-14

Copyright © 2007 Juniper Networks, Inc. Proprietary and Confidential www.juniper.net

Technical Certification Programs: Routing Tracks

This slide outlines the current levels of technical certification offered by Juniper Networks.

Technical Certification Programs: Routing Tracks

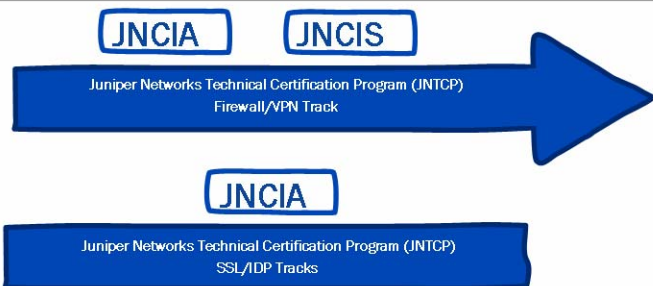


- Routing technical certification includes the following tracks:
 - Service provider track: JUNOS platforms
 - Service provider track: JUNOS platforms
 - Enterprise routing track
- Routing tracks consist of written and lab-based examination

Technical Certification Programs: Security Tracks

This slide outlines the current levels of technical certification offered by Juniper Networks.

Technical Certification Programs: Security Tracks



- Security technical certification includes the following tracks:
 - Firewall/VPN track
 - SSL/IDP tracks
- Security certification programs are written examination only at this time

Juniper your Net

Copyright © 2007 Juniper Networks, Inc. Proprietary and Confidential www.juniper.net 1-16

The JNCIA Certification

This slide details the JNCIA certification level.

Technical Certification Programs: WX Track

JNCIA

Juniper Networks Technical Certification Program (JNTCP)
WX Track

- The WX certification program is written examination only at this time



The JNCIS Certification

This slide details the JNCIS certification level.

Juniper Networks Certified Internet Associate (JNCIA)

- Computer-based, written exam
- Delivered at Prometric testing centers worldwide
- 60 questions, 60 minutes
- Passing Score: 70%
- \$125 USD
- Prerequisite certification: none
- Benefits provided to JNCIAs:
 - Certificate
 - Logo usage
 - Industry recognition
- Validates candidate's general knowledge of IP technologies, platform operating system, and hardware



The JNCIP Certification

This slide details the JNCIP certification level.

Juniper Networks Certified Internet Specialist (JNCIS)

- Computer-based, written exam
- Delivered at Prometric testing centers worldwide
- Prerequisite for the JNCIP lab exam
- 75 questions, 90 minutes
- Passing Score: 70%
- \$125 USD
- Prerequisite certification: none
- Benefits provided to JNCISs:
 - Certificate
 - Logo usage
 - Provides ability to take JNCIP exam
 - Industry recognition as an IP and routing platform specialist
- Validates candidate's advanced knowledge of platform operating system, hardware, and IP technologies



The JNCIE Certification

This slide details the JNCIE certification level.

Juniper Networks Certified Internet Professional (JNCIP)

- One-day, lab-based exam
- Tests candidate's configuration and design skills for essential technologies
- Testing centers: Sunnyvale, Amsterdam, Herndon, Westford, Remote
- Prerequisite for the JNCIE lab exam
- \$1,250 USD
- Prerequisite certification: JNCIS
- Benefits provided to JNCIPs:
 - Certificate
 - Logo usage
 - Provides ability to take JNCIE exam
 - Industry recognition as an IP and routing platform professional
- Validates candidate's practical platform configuration skills



Prepping and Studying

This slide lists some options for those interested in prepping for Juniper Networks certification.

Juniper Networks Certified Internet Expert (JNCIE)

- One-day, lab-based exam
- Tests candidate's advanced configuration and design skills for essential and specialized technologies
- Testing centers: Sunnyvale, Amsterdam, Herndon, remote
- \$1,250 USD
- Prerequisite certification: JNCIP
- Currently only available in the M-series routers track
- Benefits provided to JNCIEs:
 - Crystal plaque and certificate
 - Logo usage
 - Worldwide recognition as an Internet Expert
- The most challenging and respected exam of its type in the industry



Any Questions?

If you have any questions or concerns about the class you are attending, we suggest that you voice them now so that your instructor can best address your needs during class.



Operating Juniper Networks Routers in the Enterprise

Chapter 2: Juniper Networks Enterprise Routers

Chapter Objectives

- After successfully completing this chapter, you will be able to:
 - Describe Juniper Networks platforms targeted at the customer edge and enterprise markets
 - Describe the design architecture of Juniper Networks routers
 - List and describe Juniper Networks router components
 - Describe packet flow through a J-series platform
 - Describe interface support and naming conventions
 - List some FRUs
 - List management options for Juniper Networks enterprise routers



This Chapter Discusses:

- Juniper Networks, Inc. enterprise products and their typical applications;
- General platform architecture;
- Juniper Networks router components;
- Packet flow;
- Interface support and naming conventions;
- Some field-replaceable units (FRUs); and
- Management options.

Agenda: Juniper Networks Enterprise Routers

- Customer Edge and Enterprise Platforms
- Overview of Enterprise Routing Platforms
- Architecture and Packet Flow
- Model Specifics
- Interface Support and Naming
- Field-Replaceable Units
- Network Management Options



Customer Edge and Enterprise Platforms

The slide lists the topics we cover in this chapter. We discuss the highlighted topic first.

Juniper Networks Enterprise Routers

- Router platforms designed to meet the needs of small and large enterprises
 - Designed to scale in multiple dimensions
 - Market-leading port density
 - Flexible and manageable traffic control
 - High-reliability features
 - Value-added services



M10i

The M7i, M10i, and M120 platforms are well suited to edge services for large enterprises or for core/data center applications



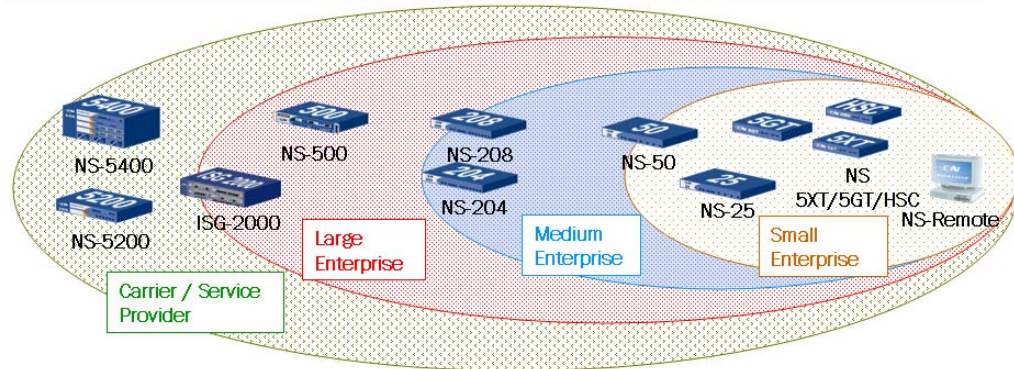
J6350

The J-series Services Router family is specifically designed for smaller enterprise environments including remote, branch, and regional offices

Juniper Networks Enterprise Product Offerings

This slide outlines the Juniper Networks routers aimed at the enterprise market. These routers offer differing combinations of price, performance, and redundancy to match the needs of both small and large enterprises.

Juniper Networks Security Platforms

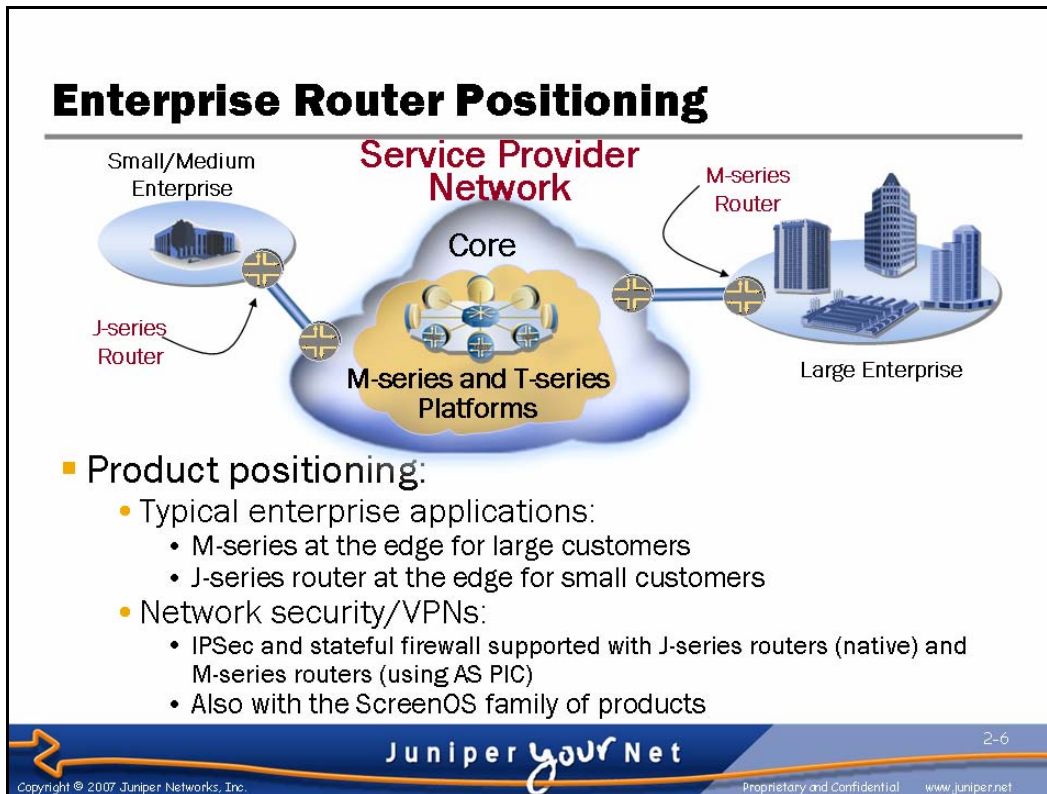


- The family of security appliances acquired as part of the NetScreen acquisition
 - Can be paired with M-series and J-series routers as part of an end-to-end solution
 - NetScreen products are covered in a separate curriculum

NetScreen Security Appliances

NetScreen security appliances offer the following benefits:

- Strong security for access control, user authentication, and network and application-level attack protection;
- Lower capital investment, and support, deployment, and operational costs for overall lower total cost of ownership (TCO); and
- Predictable performance for a highly reliable, available, and secure network.



Juniper Networks Product Positioning

The network of today's service providers is typically made up of two major components: the network edge and the network core. These two components operate differently and have different network device requirements and application focuses.

The service provider's network edge is normally associated with a large number of broadband remote access servers (B-RAS) that support large numbers of low- to medium-speed customer devices. B-RAS and other customer aggregation devices must support a variety of physical link layer technologies, such as DSL, ATM, Frame Relay, Ethernet, and dedicated access links based on T1/E1 and T3/E3 technology. Edge devices often rely on simple static routing and might provide security and class-of-service (CoS) features as needed. These network edge applications are normally served by M-series and E-series routing platforms.

In contrast, the service provider's network core is often associated with a smaller number of routers supporting far fewer interfaces that operate at much higher speed. These high-speed interfaces are typically based on SONET technology and act to aggregate the data from large numbers of individual subscriber lines for efficient long-haul transport. Core routers almost always run dynamic routing protocols, both for internal routing (IGP) and external routing (BGP), and might also deploy Multiprotocol Label Switching (MPLS) for traffic engineering and VPN-related applications. Core routers might also provide CoS, and in some cases, security-related features. Network core applications are normally served by M-series and T-series routing platforms.

Continued on next page.

Juniper Networks Product Positioning (contd.)

Enterprise customer premise applications are served by the J-series family of edge routers and, in the case of larger enterprises, M-series routers. Enterprise data center applications can also be served by M-series routers. The J-series and M-series routers support the rich security and class-of-service features needed by the enterprise while still maintaining value, stability, and predictably high performance.

Agenda: **Juniper Networks Enterprise Routers**

- Customer Edge and Enterprise Platforms
- ➔ **Overview of Enterprise Routing Platforms**
- Architecture and Packet Flow
- Model Specifics
- Interface Support and Naming
- Field-Replaceable Units
- Network Management Options



Overview of Enterprise Routing Platforms

This slide highlights the topic we cover next.

M-series Overview

- Runs industry-proven JUNOS software
- Hardware-based forwarding
 - ASIC-driven data paths provide predictable and stable performance
 - Separation of control and forwarding planes protects data plane from control-plane instability
- Ease-of-use features designed for the enterprise
 - Web-based GUI management available
 - Rescue configuration



JUNOS Software

M-series routers run the same JUNOS software that has been proven stable in the world's largest service provider networks. The JUNOS software is feature rich including the stateful firewall, VPN, and CoS functionality that is important to the enterprise environment.

Hardware-Based Control and Forwarding

M-series routers have specialized application-specific integrated circuits (ASICs) that implement their main forwarding functionality in hardware. This hardware is separate from the hardware that provides the control plane, ensuring that routing protocols and other control plane processes do not interfere with packet forwarding. This design provides predictable forwarding performance, even when you enable features and services.

Enterprise Features

You can now install the J-Web interface and use this Web-based GUI to manage M-series routers. Additionally, the JUNOS software allows you to save a rescue configuration, which you can load with the **rollback rescue** configuration command.

J-series Overview

- Runs industry-proven JUNOS software with integrated services
- Software-based control and forwarding
 - Departure from the ASIC-driven data paths in the M-series and T-series platforms
 - Software-only design keeps costs low
 - Software-only design allows greater feature flexibility
- Ease-of-use features designed for the enterprise
 - Web-based GUI management installed by default
 - Rescue configuration loadable by pushing a button
 - Autoinstallation



JUNOS Software

J-series routers run the same JUNOS software that has been proven stable in the world's largest service provider networks. The JUNOS software is feature rich including the stateful firewall, VPN, and CoS functionality that is important to the enterprise environment.

Software-Based Control and Forwarding

J-series routers, unlike the M-series and T-series, do not depend upon specialized ASIC hardware to implement their main forwarding functionality. Predictable forwarding performance is still maintained by using a real-time operating system that ensures that packet forwarding processes are given the highest priority level. This forwarding performance is maintained even when services are enabled.

Enterprise Features

The J-Web interface is installed by default on all J-series routers. Additionally, the JUNOS software supports autoinstallation and allows you to save a rescue configuration, which you load by pushing a button on the front of the router. Juniper Networks added these autoinstallation and rescue configuration features to the JUNOS software to ease the support overhead in remote locations that might not have full-time networking staff on site.

JUNOS Software

The diagram illustrates the JUNOS Software architecture. On the left, a stack of five colored blocks represents the modular operating system components: Protocols(RPD) in blue, PPPMD(Hellos) in purple, Chassis Mgmt in green, SNMP in yellow, and Interface mgt in orange. These blocks sit on a dark blue base labeled 'Operating System'. To the right, a horizontal line connects a J2300 router to a TX Matrix switch. Below this line is a box labeled 'JUNOS' containing three smaller boxes with versions 8.0, 8.1, and 8.2, connected by arrows, indicating a software train. A red text label below the diagram states: 'A single SW train runs on all platforms!'.

- Robust, modular OS with industry-leading performance and scalability
 - The next-generation operating system, since it was introduced!
- Separates forwarding and control planes for maximum stability and reliability
- A single software train for J-series, M-series, and T-series platforms

Juniper your Net

Copyright © 2007 Juniper Networks, Inc. Proprietary and Confidential www.juniper.net

Robust, Modular, and Scalable

JUNOS software functionality is compartmentalized into multiple software processes that each handle a portion of the router's functionality. Each process runs in its own protected memory space, ensuring that one process cannot directly interfere with another. When a single process fails, the entire routing system does not necessarily fail. This modularity also ensures that new features can be added with less likelihood of breaking current functionality.

Separate Forwarding and Control Planes

Another aspect of the JUNOS software's modularity is the separation of forwarding and control planes. The processes that control routing protocols are cleanly separated from the processes that forward packets through the router. This design allows each process to be tuned for maximum performance and reliability.

Single Software Source Code

JUNOS software on the J-series platform uses the same source code as on the M-series and T-series platforms. This design ensures that features work the same across every platform—from the J2300 Services Router to the TX Matrix. Enabling new software features does not require changing to a different JUNOS binary. JUNOS software costs are kept low by a soft-licensing model that ensures J-series router customers do not pay for unused features.

M-series Platform Components

- M-series platforms comprise custom hardware
 - Routing Engine runs on an x86 architecture microprocessor
 - Compact flash device and hard drive for mass storage
 - Forwarding plane (PICs, FPCs, FEBs, SIBs, etc.) uses custom ASICs to perform forwarding
 - All forwarding performed in hardware
 - Physical interfaces provided by PICs
 - Great flexibility in mixing different interface types in a single chassis

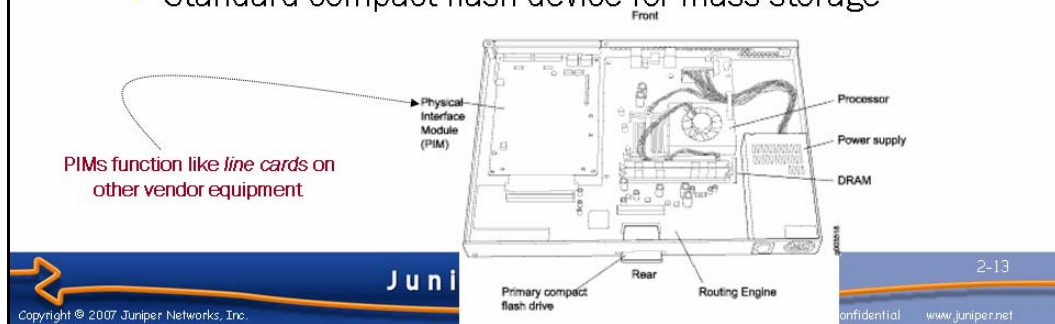


Custom Hardware

M-series routers use custom, purpose-built hardware to provide exceptional and consistent performance. The Routing Engine (RE) runs on an x86 architecture microprocessor and provides control-plane functions, while the separate forwarding plane uses custom ASICs to perform forwarding. Juniper Networks manufactures a wide variety of PICs that provide different interfaces. You can mix different PICs within a Flexible PIC Concentrator (FPC) (somewhat analogous to other vendors' line cards) or chassis to provide the right mixture of interfaces for your environment. PICs are generally reusable across most M-series platforms. There are some restrictions, which are noted in the official documentation available on the Juniper Networks Web site.

J-series Platform Components

- J-series platforms comprise standard PC components to help reduce cost
 - Routing and Forwarding Engine runs on an x86 architecture microprocessor
 - Intel IXP4xx family of network processors provide PCI-based WAN/LAN interfaces
 - Scalable design results in increased processing power with each interface addition!
 - Standard compact flash device for mass storage



Commodity Hardware

The use of standard PC components provides excellent performance at a reasonable price, thanks to the benefit of volume pricing. An x86 architecture processor handles the RE and Packet Forwarding Engine (PFE) functionality, while Intel IXP4xx network processors offer a standard and low-cost mechanism for handling network-specific functionality in a scalable fashion.

J-series Copyright Protection

- Copyright protection exists to prevent running JUNOS software on nonlicensed hardware
 - J-series motherboards contain an EPROM signed by the Juniper Networks private key
 - Also contains a device ID that can be tied to support contracts and feature licensing
 - Can contain a set of unique default passwords for remote and automatic provisioning
 - Unique serial number prevents copying an existing EPROM
 - JUNOS software is signed by a corresponding public key
- Copyright violation results in the termination of the chassis manager
 - No forwarding



J-series Copyright Protection

While the J-series router consists primarily of standard PC hardware, this design does not mean you can run the JUNOS software on a PC. Juniper Networks uses public-key cryptography to provide copyright protection.

No Forwarding

JUNOS software will not forward packets unless it can verify that the EPROM is properly signed by the Juniper Networks private key. This copyright protection ensures that the JUNOS software functions only on supported Juniper Networks hardware.

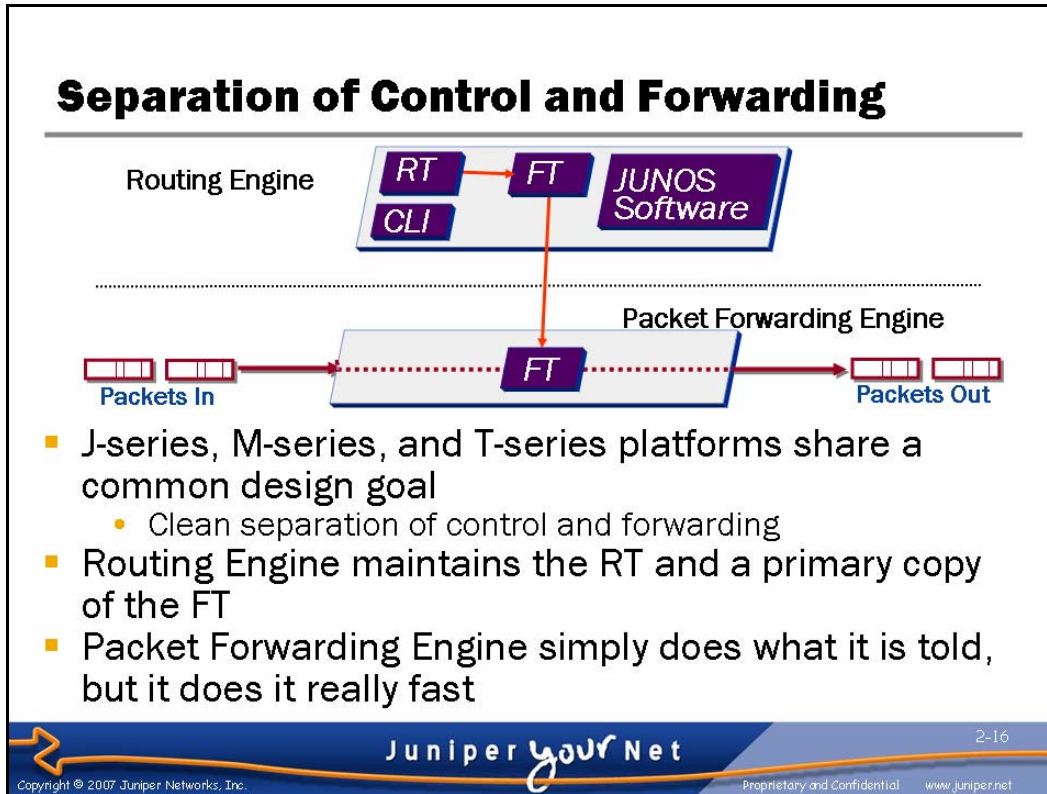
Agenda: Juniper Networks Enterprise Routers

- Customer Edge and Enterprise Platforms
- Overview of Enterprise Routing Platforms
- **Architecture and Packet Flow**
- Model Specifics
- Interface Support and Naming
- Field-Replaceable Units
- Network Management Options



Architecture and Packet Flow

This slide highlights the topic we discuss next.



Architectural Philosophy

Architecturally, all Juniper Networks J-series, M-series, and T-series platforms share a common design that separates the router's control and forwarding planes. To this end, all J-series, M-series and T-series platforms consist of two major components:

- *The Routing Engine (RE):* The RE is the brains of the platform; it is responsible for performing routing updates and system management. The RE runs various protocol and management software processes that live inside a protected memory environment. The RE is a general-purpose computer platform based on an x86 architecture microprocessor. The RE maintains the router's primary forwarding table and is connected to the PFE through an internal link.
- *The Packet Forwarding Engine (PFE):* The PFE is responsible for forwarding transit packets through the router. The PFE is implemented using real-time threads on J-series platforms and with ASICs on the M-series and T-series platforms. Because this architecture separates control operations—such as routing updates and system management—from packet forwarding, the router can deliver superior performance and highly reliable deterministic operation, even in the case of the software-based J-series PFE.

Continued on next page.

Routing and Forwarding Table Interaction

The JUNOS software routing protocol process implements the various routing protocols that can be run on the router. The routing protocol process starts all configured routing protocols and handles all routing messages. The routing process maintains one or more routing tables, which consolidates the routing information learned from all routing protocols into common tables. From this routing information, the routing protocol process determines the active routes to network destinations and installs these routes into the RE's forwarding table (FT).

The PFE receives the forwarding table from the RE via an internal link. FT updates are a high priority for the JUNOS software kernel and are performed incrementally. Entries are never aged out of the FT to make room for new entries or because they have not been recently used. This design ensures consistently high forwarding performance for every packet in every network flow to every network destination.

The PFE Does What It Is Told

Because the RE provides the intelligence side of the equation, the PFE can simply do what it is told to do—that is, forward packets with a high degree of stability and deterministic performance.

The Routing Engine

- Maintains the routing and forwarding tables used by the router
 - Based on one or more real-time operating system threads
 - Provides forwarding tables to the PFE
- Controls and monitors the chassis
 - Implements the command-line and network management interfaces
 - Provides power control and system status monitoring
- Manages PFE



Routing Engine Intelligence

The RE handles all the routing protocol processes as well as other software processes that control the router's interfaces, the chassis components, system management, and user access to the router. These routing and software processes run on top of the JUNOS kernel that interacts with the PFE. All routing protocol packets from the network are directed to the RE.

Controls and Monitors

The RE provides the command-line interface (CLI) as well as the J-Web graphical user interface (GUI). These user interfaces run on top of the JUNOS kernel and provide user access and control of the router. We provide a detailed examination of JUNOS user interfaces and their features in a subsequent chapter.

Packet Forwarding Engine Management

The RE controls the PFE by providing an accurate and up-to-date forwarding table and by downloading microcode and managing software processes that live in the PFE's microcode. The RE receives hardware and environmental status messages from the PFE and acts upon them as appropriate.

The Packet Forwarding Engine— M-series and T-series Platforms

- Custom ASICs implement forwarding path
 - No *process switching*
 - Value-added services and features implemented in hardware
 - Multicast
 - CoS/queuing
 - Firewall filtering
 - Accounting
- Divide-and-conquer architecture
 - Each ASIC provides a piece of the forwarding puzzle

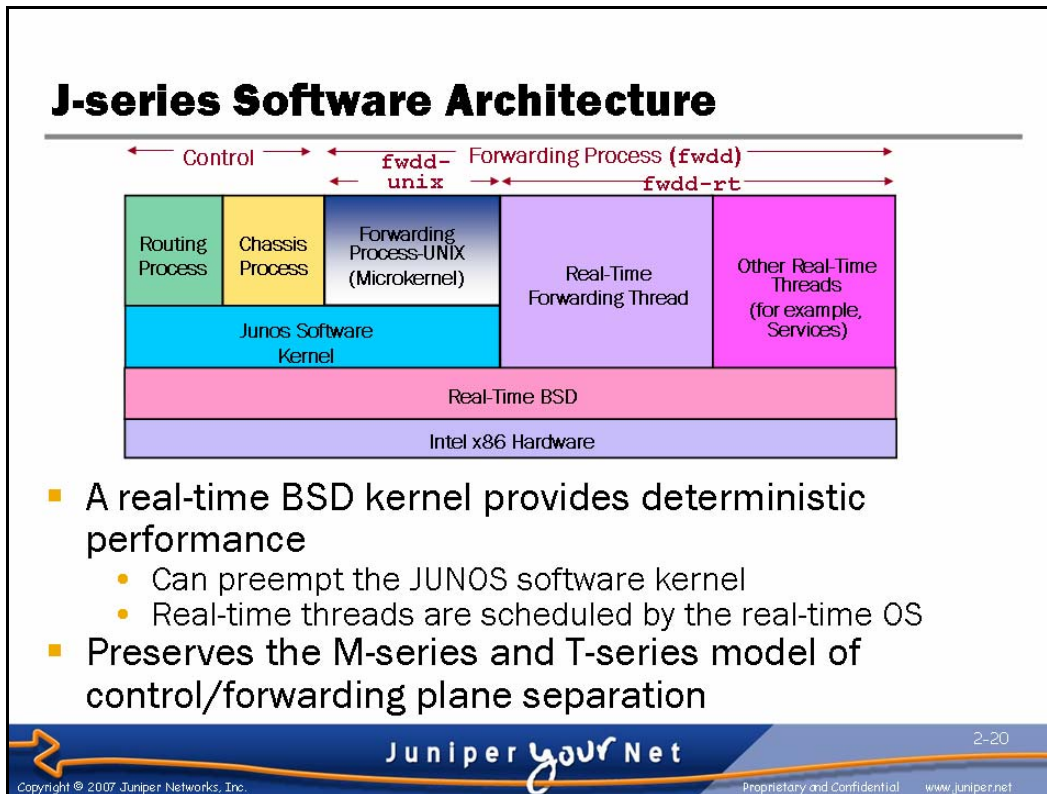


Custom ASICs

ASICs enable the router to achieve data forwarding rates that match current fiber-optic capacity. Such high forwarding rates are achieved by distributing packet processing tasks across highly integrated ASICs. As a result, Juniper Networks M-series and T-series platforms do not require a general purpose processor for packet forwarding; this makes *process switching* (the software-based handling of packet forwarding) an alien concept for Juniper Networks routers. The custom ASICs provide enhanced services and features, such as multicast, CoS/queuing, and firewall filtering in hardware so that you can enable services on production routers without concern of significant performance hits.

Divide-and-Conquer Architecture

Each ASIC provides a piece of the forwarding puzzle, allowing a single ASIC to perform its specific task optimally. These ASICs work together to consistently forward each packet at wire-rate speeds, while performing the tasks you configure.



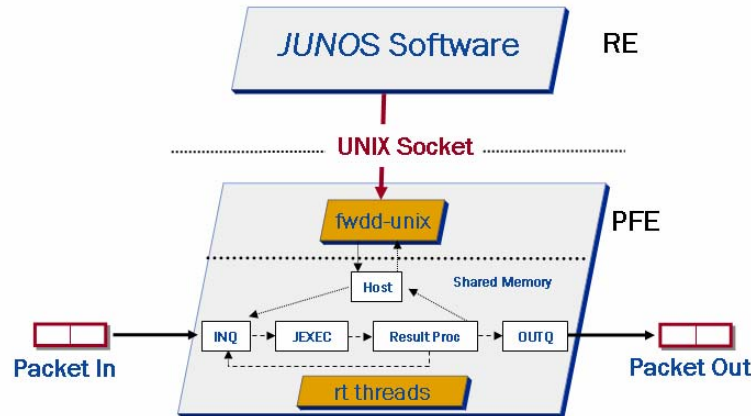
Deterministic Performance

The J-series RE and software PFE are both implemented on the primary x86 architecture microprocessor. A real-time operating system kernel mediates access to the underlying hardware. The real-time kernel ensures that operating system services are delivered in a constant, load-independent, amount of time. This design ensures that the forwarding and services real-time threads deliver predictable packet forwarding performance.

Control and Forwarding Separation

Logical separation between the control and forwarding planes is maintained by separate real-time processes. Control plane processes continue to run on the traditional JUNOS software kernel that is a client of the real-time kernel. Forwarding and services threads run directly on the real-time kernel.

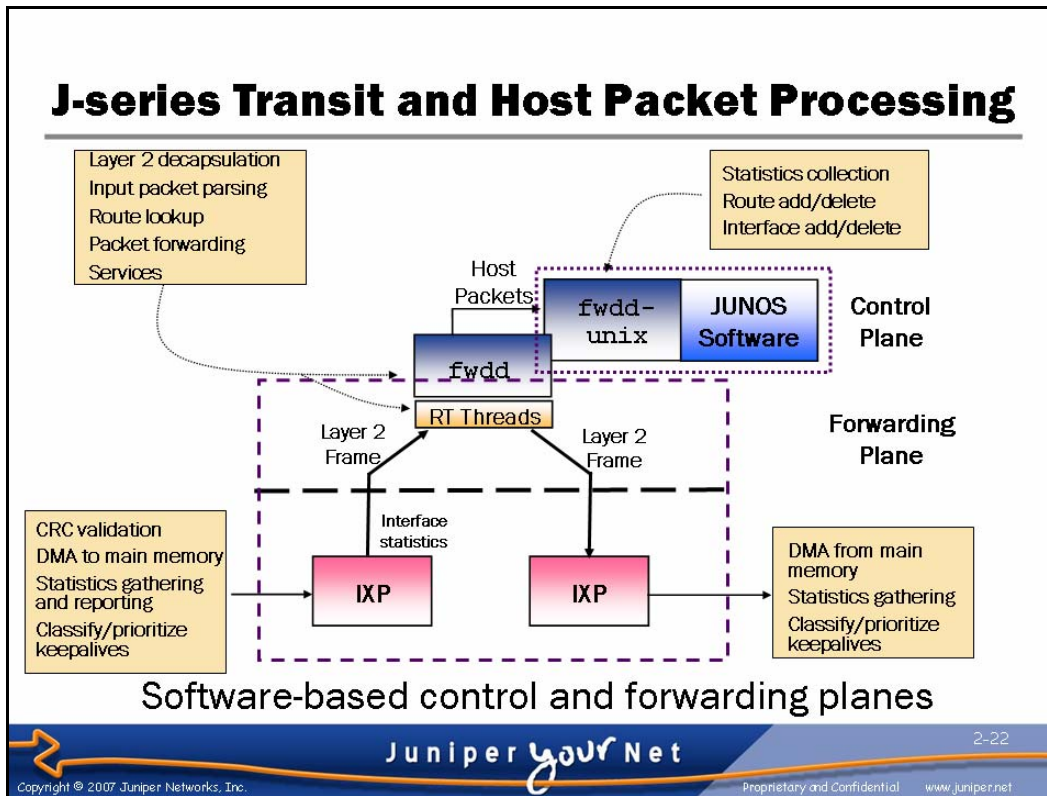
The Net Result: A Virtualized PFE



- The `fwdd-unix` process emulates the microkernel found in M-series or T-series PFE
 - Allows use of existing APIs between the RE/JUNOS software and the forwarding process
 - RE communicates to `fwdd-unix` through a UNIX socket

J-series Virtual Packet Forwarding Engine

The J-series software PFE maintains at a fraction of the cost many of the benefits of the microkernel and ASIC-based PFE found on the M-series and T-series platforms. A UNIX socket provides the internal link between the RE and PFE and allows the JUNOS control plane software from the M-series and T-series platforms to be reused on the J-series platform.



Packet Processing

While the virtual PFE handles packet forwarding decisions in software, the Intel IXP network processors still provide performance scalability. These network processors handle Layer 2 functions such as cyclic redundancy check (CRC) validation, statistics gathering, classification, and keepalives. Because an IXP processor is on each Physical Interface Module (PIM), overall router capability increases as PIMs are added.

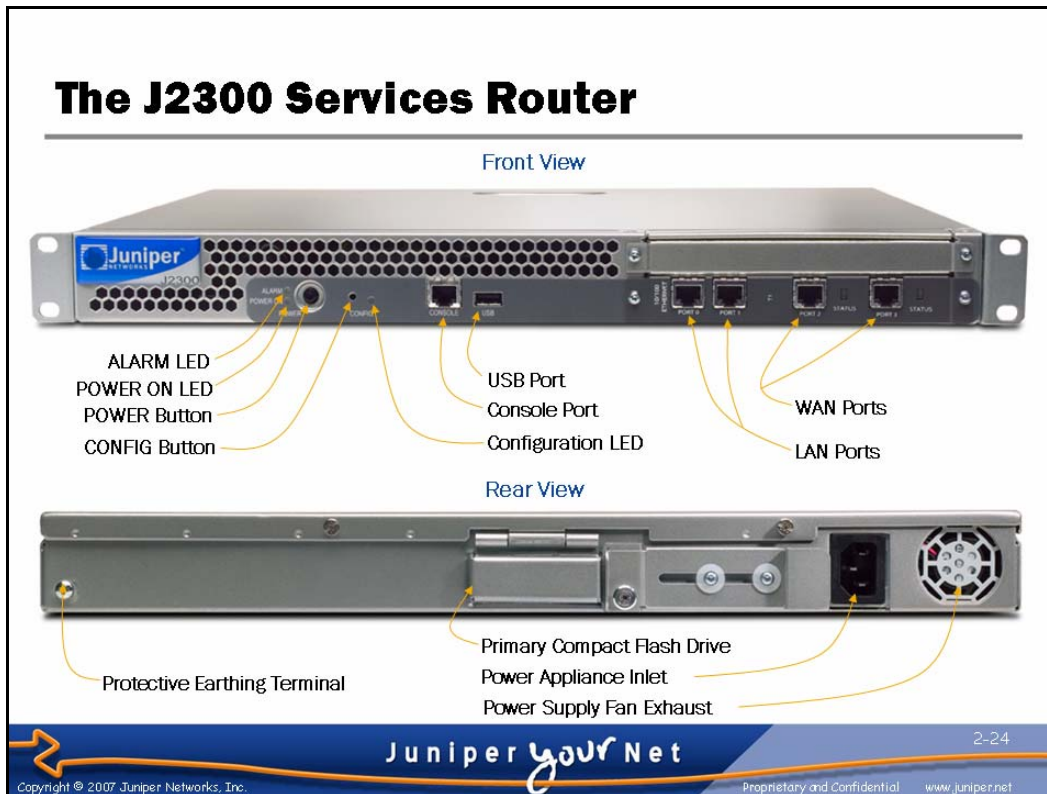
Agenda: Juniper Networks Enterprise Routers

- Customer Edge and Enterprise Platforms
- Overview of Enterprise Routing Platforms
- Architecture and Packet Flow
- Model Specifics
 - Interface Support and Naming
 - Field-Replaceable Units
 - Network Management Options



Model Specifics

This slide highlights the topic we cover next.



The J2300 Services Router

The J2300 Services Router is the entry level J-series platform. It is ideal for remote office locations that are connected using one or two T1/E1 circuits. The J2300 platform has the following features:

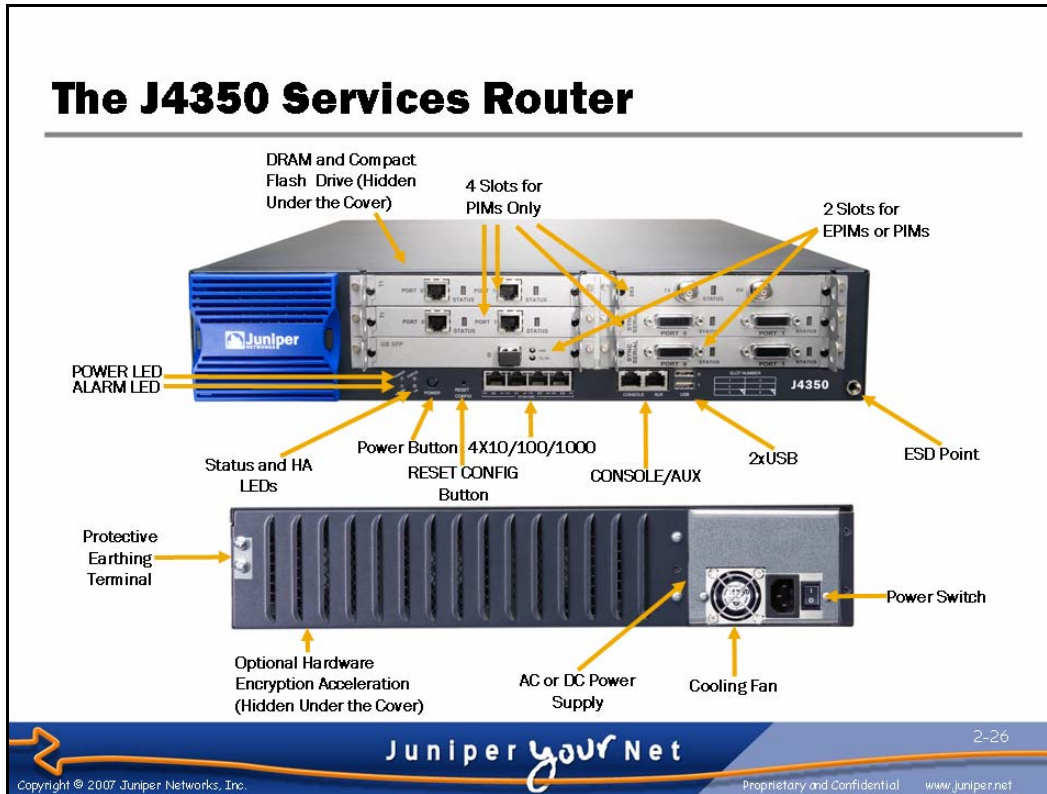
- Compact 1-RU form factor;
- 256-MB DRAM, field-upgradeable to 512 MB;
- 128-MB primary compact flash disk, field-upgradeable to 1 GB;
- USB port that allows USB flash drives to be used as secondary device;
- Designed for 8 Mbps of throughput;
- A variety of fixed WAN interface

Continued on next page.

The J2300 Services Router (contd.)

The J2300 Services Router includes the following front and rear panel components:

- *ALARM LED*: This yellow (amber) LED lights to indicate a critical condition that can result in a system shutdown or a less severe condition that requires monitoring or maintenance. This is a single-color alarm, regardless of the severity of the alarm condition (critical, major, or minor).
- *POWER ON LED*: This green LED is off when the router is unplugged or is powered off and in standby mode. It lights steadily when the router is powered on and is either booting or functioning normally. The *POWER ON LED* blinks when the router is gracefully shutting down.
- *Power button*: Pressing and releasing the power button will power on a J-series Services Router that is currently powered off. Briefly press and release the power button to initiate a graceful shutdown and power off a running router. Pressing the power button for more than 5 seconds will immediately power off the router. Perform this method only after gracefully shutting down the operating system from the user interface.
- *CONFIG button*: This recessed button performs two recovery operations. Press and release it to load and commit the user-defined rescue configuration. Press and hold the CONFIG button for at least 15 seconds to delete all configurations, and then load and commit the factory-default configuration. You can disable either or both of these capabilities in the configuration.
- *Configuration LED*: This LED blinks green while the rescue configuration is being loaded. It lights steadily green when the rescue configuration or factory-default configuration is loaded and committed. The configuration LED blinks red while all configurations are being deleted and the factory-default configuration is being loaded and committed. The configuration LED lights steadily red if a recovery operation fails.
- *Console port*: This port is a data terminal equipment (DTE) RS-232 serial port with RJ-45 connector used to access the router's CLI.
- *USB port*: This port is a universal serial bus (USB) port that accepts a USB storage device for use as a secondary storage device.
- *LAN ports*: These ports are two fixed 10/100 Base-TX Fast Ethernet ports.
- *WAN Ports*: These ports are two fixed T1, E1, or synchronous serial ports.
- *Protective earthing terminal*: This terminal is the attachment point for a grounding cable that connects the router to earth ground.
- *Primary compact flash drive*: This drive provides primary storage for log files, configuration files, and software images.
- *Power appliance inlet*: This inlet is the attachment point for the fixed AC or DC power cord. Power cords are available with plugs appropriate for each geographical location.
- *Power supply fan exhaust*: This is the exhaust for the cooling fan on the router's autosensing power supply.



The J4350 Services Router

The J4350 Services Router is the mid-level J-series platform. It is ideal for remote office locations that are connected using n x T1/E1 circuits. The J4350 platform has the following features:

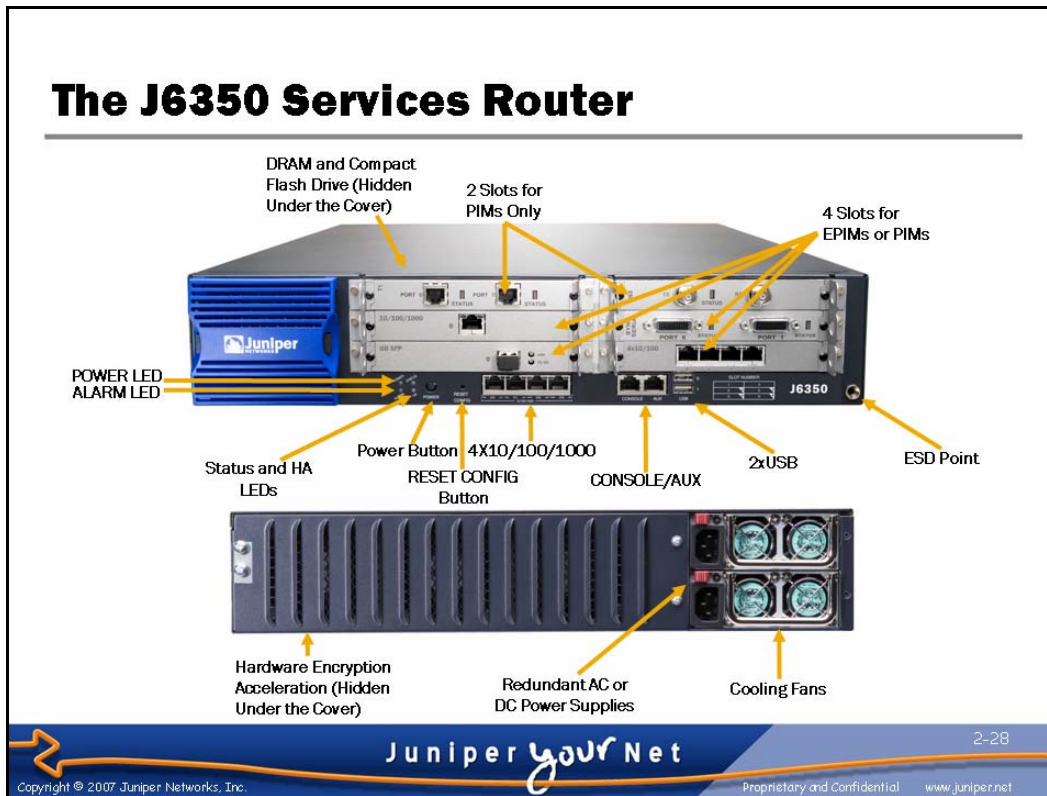
- 2-RU form factor;
- 256-MB or 1-GB DRAM, field-upgradeable to 2 GB;
- 256-MB primary compact flash drive, field-upgradeable to 1 GB;
- Two USB ports that allow USB flash drives to be used as secondary devices;
- 600-Mbps clear-text throughput (with the IMIX test);
- 30-Mbps encrypted (3DES+SHA1 IPSec VPN) throughput without optional hardware acceleration or 300-Mbps encrypted throughput with optional hardware acceleration;
- Modular design with four built-in 10/100/1000-Mbps Ethernet ports, two Enhanced Physical Interface Module (EPIM)/PIM slots, and four PIM-only slots; and
- PIMs available in many configurations.

Continued on next page.

The J4350 Services Router (contd.)

The J4350 Services Router includes the following front and rear panel components:

- *ESD point*: This is a banana plug receptacle for attaching an ESD wrist strap.
- *ALARM LED*: This LED lights red to indicate a critical condition that can result in a system shutdown or yellow to indicate a less severe condition that requires monitoring or maintenance.
- *POWER LED*: This green LED is off when the router is unplugged or is powered off and in standby mode. It lights steadily when the router is powered on and is either booting or functioning normally. The *POWER LED* blinks when the router is gracefully shutting down.
- *Power button*: Pressing and releasing the power button will power on a J-series Services Router that is currently powered off. Briefly press and release the power button to initiate a graceful shutdown and power off a running router. Pressing the power button for more than 5 seconds will immediately power off the router. Perform this method only after gracefully shutting down the operating system from the user interface.
- *RESET CONFIG button*: This recessed button performs two recovery operations. Press and release it to load and commit the user-defined rescue configuration. Press and hold the *RESET CONFIG* button for at least 15 seconds to delete all configurations, and then load and commit the factory-default configuration. You can disable either or both of these capabilities in the configuration.
- *Status LED*: This LED blinks green while the router is starting up or performing diagnostics. It lights steadily green when the router is booted and operating normally. The configuration LED blinks red when an error is detected.
- *High Availability (HA) LED*: This LED is reserved for future use and should not be lit with this software version.
- *Console port*: This port is a DTE RS-232 serial port with RJ-45 connector used to access the router's CLI.
- *USB ports*: These ports are two USB ports that accept a USB storage device for use as a secondary storage device.
- *LAN ports*: These ports are four fixed 10/100/1000 Base-TX Fast Ethernet ports.
- *Protective earthing terminal*: This terminal is the attachment point for a grounding cable that connects the router to earth ground.
- *Power appliance inlet*: This inlet is the attachment point for the DC power leads or removable AC power cord. AC power cords are available with plugs appropriate for each geographical location.
- *Power supply fan exhaust*: This is the exhaust for the cooling fan on the router's autosensing power supply.



The J6350 Services Router

The J6350 Services Router is the top-level J-series platform. It is ideal for remote office locations that are connected using one or two DS3 circuits. The J6350 platform has the following features:

- 2-RU form factor;
- 1-GB DRAM, field-upgradeable to 2 GB;
- 256 MB primary compact flash drive, field-upgradeable to 1 GB;
- Two USB ports that allow USB flash drives to be used as secondary devices;
- 1-Gbps clear-text throughput (with the IMIX test);
- 500-Mbps encrypted (3DES+SHA1 IPSec VPN) throughput with standard hardware acceleration;
- Modular design with four built-in 10/100/1000-Mbps Ethernet ports, four EPIM/PIM slots, and two PIM-only slots;
- Redundant (hot-swappable) AC or DC power supply option; and
- PIMs available in many configurations.

The J6350 router includes the same front and rear panel components as the J4350 router.

J-series RE Characteristics

J-series Model

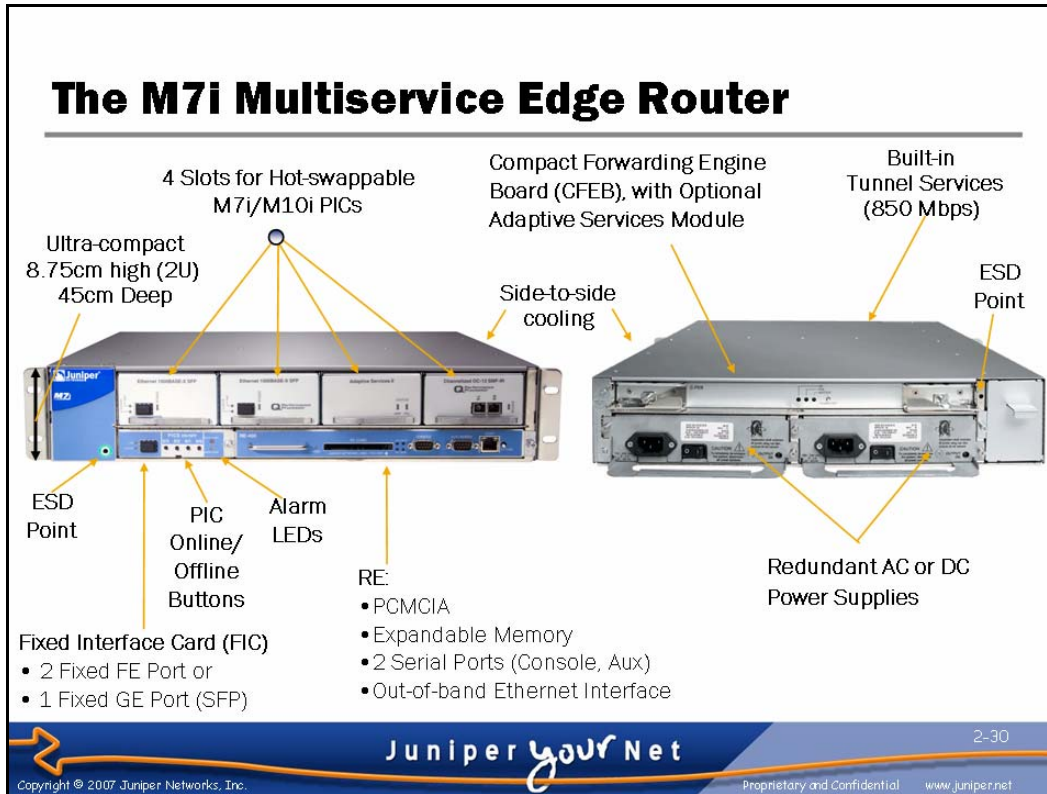
Feature	J6350	J4350	J2300
Processor/clock	Pentium IV 3.4 GHz	Celeron-D 2.53 GHz	Celeron 1.6 GHz
Memory (Min/Max)	1 GB / 2 GB (Non-ECC)	256 MB / 2 GB (Non-ECC)	256 MB / 512 MB (Non-ECC)
Compact Flash (Min/Max)	256 MB / 1 GB	256 MB / 1 GB	128 MB
HW Encryption Acceleration	Yes	Optional	Not Supported
USB Support (1.1 or greater)	2 USB Ports	2 USB Ports	1 USB Port *
Management Ports	One EIA-232 Port (RJ-45 connector)	One EIA-232 Port (RJ-45 connector)	One EIA-232 Port (RJ-45 connector)

* See release notes for applicable JUNOS software version at <http://www.juniper.net/techpubs/software/jseries/> for a list of supported USB devices.



RE Comparison: J-series Models

This slide provides a matrix of key characteristics associated with J-series REs.



The M7i Multiservice Edge Router

The M7i Multiservice Edge Router is a compact M-series router. It is ideal for locations requiring the performance and flexible interface configurations of an M-series box in a small form factor. The M7i router has the following features:

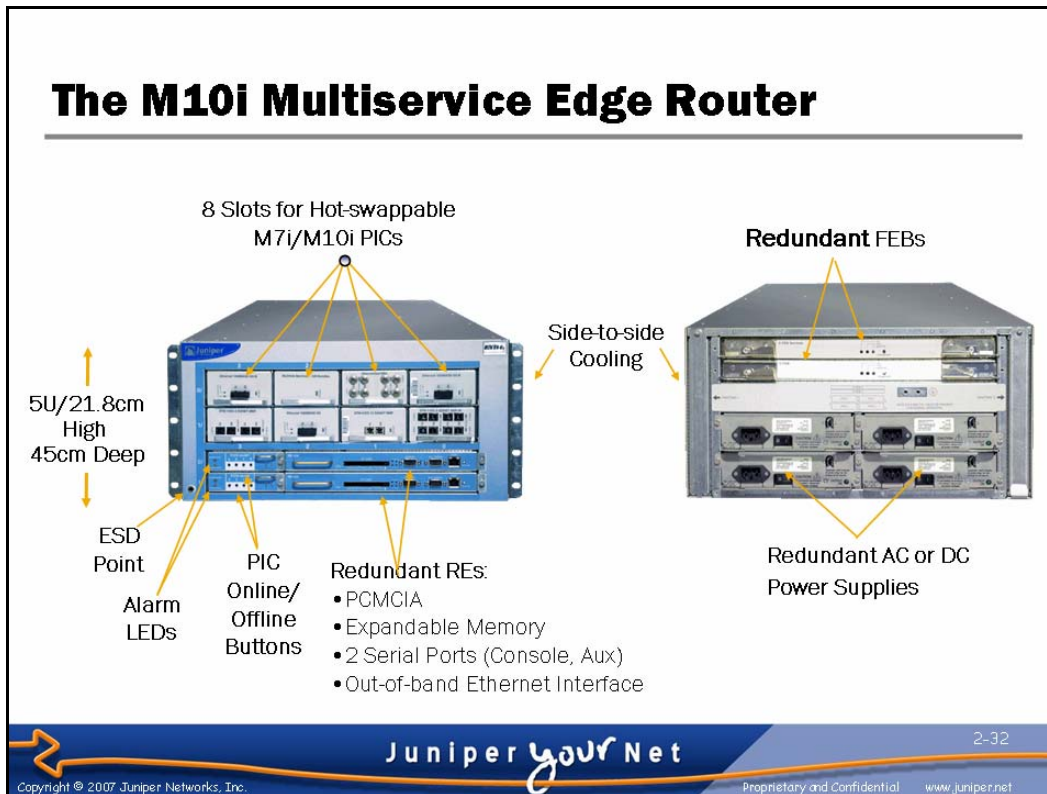
- 2-RU form factor;
- Storage on compact flash drive and hard drive;
- 8.4-Gbps (16-Mpps) cleartext throughput;
- Integrated tunnel services module;
- Optional integrated Adaptive Services Module (ASM) with support for Layer 2 and Layer 3 features (MLPPP, MLFR, CRTP, stateful firewall, NAT, IDS, and IPSec VPN), supporting 256-Mbps (100-Kpps) cleartext throughput or 200-Mbps (100-Kpps) encrypted throughput;
- Available Adaptive Services (AS) PIC with support for Layer 2 and Layer 3 features (MLPPP, MLFR, CRTP, stateful firewall, NAT, IDS, and IPSec VPN), supporting 800-Mbps (400-Kpps) cleartext throughput or 640-Mbps (312-Kpps) encrypted throughput;
- Modular design with two built-in Fast Ethernet ports or one built-in Gigabit Ethernet port and four PIC slots; and
- PICs available in many configurations.

Continued on next page.

The M7i Multiservice Edge Router (contd.)

The M7i Multiservice Edge Router includes the following front and rear panel components:

- *ESD points*: These are banana plug receptacles for attaching an ESD wrist strap.
- *ALARM LEDs*: The red LED lights to indicate a critical condition that can result in a system shutdown, while the yellow LED lights to indicate a less severe condition that requires monitoring or maintenance.
- *PIC Online/Offline buttons*: To safely remove a PIC, you must first take the PIC offline by using the JUNOS CLI or by pressing the *PIC Online/Offline* button. Before using a PIC you have inserted, you must put the PIC online by using the CLI or by pressing the *PIC Online/Offline* button.
- *Console port*: This port is a DTE RS-232 serial port with DB-9 connector used to access the router's CLI.
- *LAN ports*: The router has either two Fast Ethernet ports or one Gigabit Ethernet port on the FIC.
- *Out-of-band Fast Ethernet Management Port*: There is one out-of-band Fast Ethernet management port for each RE. This interface is identified in software as `fxp0`. This port is located on the RE.
- *RE*: The RE contains the console, modem, and out-of-band Fast Ethernet management ports. The RE also has a PCMCIA slot that can be used to install the JUNOS software. The PCMCIA slot should normally be empty.
- *Compact Forwarding Engine Board (CFEB)*: The CFEB is accessible from the rear of the chassis. If you order the optional ASM, it is integrated into the CFEB.
- *Power supplies*: The redundant power supplies are accessible from the rear of the chassis.
- *Cooling fans*: Cooling fans provide a side-to-side airflow through the router.



The M10i Multiservice Edge Router

The M10i Multiservice Edge Router is a larger M-series router. It is ideal for locations requiring the performance and flexible interface configurations of an M-series box with greater redundancy or more interfaces than the M7i router provides. The M10i router has the following features:

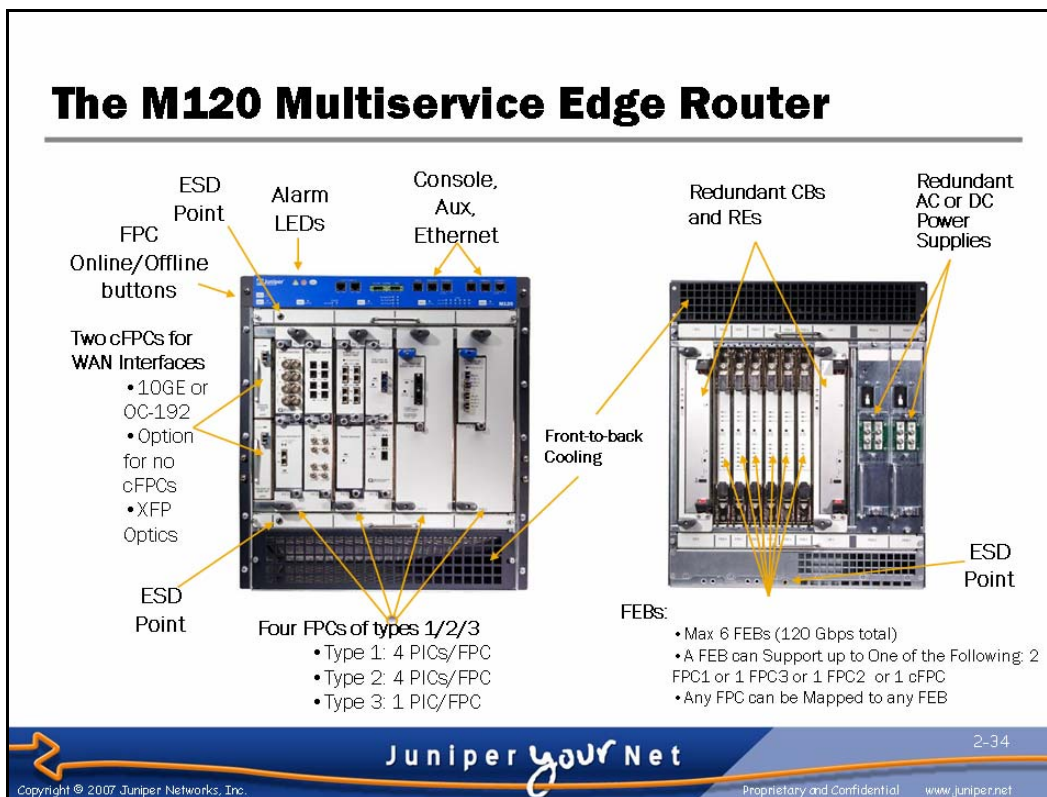
- 5-RU form factor;
- Storage on compact flash drive and hard drive;
- 12.8-Gbps (16-Mpps) cleartext throughput;
- Redundant REs, CFEs, and power supplies available;
- Available Adaptive Services (AS) PIC with support for Layer 2 and Layer 3 features (MLPPP, MLFR, CRTP, stateful firewall, NAT, IDS, and IPSec VPN), supporting 800-Mbps (400-Kpps) cleartext throughput or 640-Mbps (312-Kpps) encrypted throughput; and
- PICs available in many configurations.

Continued on next page.

The M10i Multiservice Edge Router (contd.)

The M10i Multiservice Edge Router includes the following front and rear panel components:

- *ESD points*: These are banana plug receptacles for attaching an ESD wrist strap.
- *ALARM LEDs*: The red LED lights to indicate a critical condition that can result in a system shutdown, while the yellow LED lights to indicate a less severe condition that requires monitoring or maintenance.
- *PIC Online/Offline buttons*: To safely remove a PIC, you must first take the PIC offline by using the JUNOS CLI or by pressing the *PIC Online/Offline* button. Before using a PIC you have inserted, you must put the PIC online by using the CLI or by pressing the *PIC Online/Offline* button.
- *Console port*: This port is a DTE RS-232 serial port with DB-9 connector used to access the router's CLI.
- *Out-of-band Fast Ethernet Management Port*: There is one out-of-band Fast Ethernet management port for each RE. This interface is identified in software as `fxp0`. This port is located on the RE.
- *RE*: The RE contains the console, modem, and out-of-band Fast Ethernet management ports. The RE also has a PCMCIA slot that can be used to install the JUNOS software. The PCMCIA slot should normally be empty. A *MASTER LED* next to each RE lights when the RE in that slot is the master RE.
- *CFEB*: The CFEB is accessible from the rear of the chassis. There is no option for an integrated ASM in an M10i router.
- *Power supplies*: The redundant power supplies are accessible from the rear of the chassis.
- *Cooling fans*: Cooling fans provide a side-to-side airflow through the router.



The M120 Multiservice Edge Router

The M120 Multiservice Edge Router is a larger M-series router. It is ideal for locations requiring higher interface speeds or more CoS features than the M10i router provides. The M120 router has the following features:

- 12-RU form factor;
- Storage on compact flash drive and hard drive;
- 120-Gbps (90-Mpps) cleartext throughput;
- Extensive hardware CoS support;
- Modular configuration consisting of two compact FPC (cFPC) slots and four FPC slots that accept type 1, 2, or 3 FPCs;
- Redundant REs, control boards, FEBs, and power supplies available;
- Available AS PIC with support for Layer 2 and Layer 3 features (MLPPP, MLFR, CRTP, stateful firewall, NAT, IDS, and IPSec VPN), supporting 800-Mbps (400-Kpps) cleartext throughput or 640-Mbps (312-Kpps) encrypted throughput; and
- PICs available in many configurations.

Continued on next page.


The M120 Multiservice Edge Router (contd.)

The M120 Multiservice Edge Router includes the following front and rear panel components:

- *ESD points*: These are banana plug receptacles for attaching an ESD wrist strap.
- *ALARM LEDs*: The red LED lights to indicate a critical condition that can result in a system shutdown, while the yellow LED lights to indicate a less severe condition that requires monitoring or maintenance.
- *FPC Online/Offline buttons*: To safely remove an FPC, you must first take the FPC offline by using the JUNOS CLI or by pressing the *FPC Online/Offline* button. Before using an FPC you have inserted, you must put the FPC online by using the CLI or by pressing the *FPC Online/Offline* button.
- *PIC Online/Offline buttons*: On type 1 FPCs, the *PIC Online/Offline* buttons are on the FPC next to the PIC. On type 2 and type 3 FPCs, the *PIC Online/Offline* buttons are on the PIC faceplate. To safely remove a PIC, you must first take the PIC offline by using the JUNOS CLI or by pressing the *PIC Online/Offline* button. Before using a PIC you have inserted, you must put the PIC online by using the CLI or by pressing the *PIC Online/Offline* button.
- *Console port*: This port is a DTE RS-232 serial port with RJ-45 connector used to access the router's CLI. There is one console port for each RE.
- *Out-of-band Fast Ethernet Management Port*: There is one out-of-band Fast Ethernet management port for each RE. This interface is identified in software as `fxp0`. This port is located on the front of the M120 router next to the console and modem ports.
- *RE*: The RE has a USB port that can be used for external storage.
- *CB*: The control board has an LED that indicates which CB/RE combination is the master. The control board also provides the switch fabric between the FEBs.
- *FEBs*: The FEBs are accessible from the rear of the chassis. You must install sufficient FEBs for the FPCs installed in the router. Each FEB can support two Type 1 FPCs or one Type 2, Type 3, or compact FPC. You can have extra FEBs installed for redundancy.
- *Power supplies*: The redundant power supplies are accessible from the rear of the chassis.
- *Cooling fans*: Cooling fans provide a front-to-back airflow through the router.

Feature	Platform					
	M120 Router	M10i Router	M7i Router	J6350 Router	J4350 Router	J2300 Router
Chassis Throughput (Aggregate)	120 Gbps (90 Mpps)	12.8 Gbps (16 Mpps)	8.4 Gbps (16 Mpps)	1 Gbps	600 Mbps	8 Mbps
Slot Throughput (Aggregate)	FPC1: 8 Gbps FPC2/3: 20 Gbps	6.4 Gbps (2 slots)	6.4 Gbps (1 slot)	Not Applicable	Not Applicable	Not Applicable
PICs/Ports	4 FPCs + 2 eFPCs	8 PICs	4 PICs (+2 FE or 1 GE built-in ports)	4 EPIM/PIM, 2 PIM, 4 built-in 10x100x1000	2 EPIM/PIM, 4 PIM, 4 built-in 10x100x1000	4 ports (2 FE, 2 Serial/WAN)
Power	AC/DC	AC/DC	AC/DC	AC/DC	AC/DC	AC/DC
Units per 6' Rack (70 in/1.78 m)	3	8	21	21	21	42
RE/Control Redundancy	Yes	Yes	No	No	No	No
Power Redundancy	Yes	Yes	Yes	Yes	No	No
Weight (Max)	230 Lbs/ 104.3 Kg	79 Lbs/ 36 Kg	38.2 Lbs/ 17.3 Kg	30.7 Lbs/ 13.9 Kg	25.3 Lbs/ 11.5 Kg	12 Lbs/ 5.4 Kg

* Numbers quoted are two times the unidirectional (simplex) capacity for each FPC or chassis.


Juniper your Net

Copyright © 2007 Juniper Networks, Inc.
 2-36

Proprietary and Confidential www.juniper.net

Product Comparison: M-series and J-series Platforms

This slide provides a matrix of some key characteristics of the M-series and J-series products commonly deployed in the enterprise.

Agenda: Juniper Networks Enterprise Routers

- Customer Edge and Enterprise Platforms
- Overview of Enterprise Routing Platforms
- Architecture and Packet Flow
- Model Specifics
- Interface Support and Naming
- Field-Replaceable Units
- Network Management Options



Interface Support and Naming

The slide highlights the topic we discuss next.

M-series Interface Numbering

- Based on an MM-F/P/T model, where:
 - MM = The media type (e1, fe, se, t1, t3, etc.)
 - F = The FPC number
 - P = The PIC number
 - T = The port number



Example: 0/2/1:
FPC 0
PIC 2
Port 1



M-series Interface Numbering

M-series routers use the model shown on the slide to determine physical interface designations. On the M7i router, all PICs are installed in FPC 0, while the ports on the FIC are installed on FPC 1. On all other M-series routers, the FPC designations are printed on the chassis.

J-series Interface Naming

- J-series interfaces follow the same three-level naming convention as used in M-series and T-series platforms
 - Based on an MM-F/P/T model, where:
 - MM = The media type (e1, fe, se, t1, t3, etc.)
 - F = The PIM slot number; fixed interfaces use slot 0, expansion slots numbered from left to right, bottom to top
 - P = The virtual PIM number, currently always set to 0
 - T = The port number
- Interface naming example showing a J2300 router with 2 x FE and 2 x T1 interfaces:



J-series Interface Naming

Some physical and logical media types that are currently supported on the J-series platform include the following:

- at: ADSL interface;
- dsc: Virtual interface that discards packets;
- e1: E1 WAN interface;
- fe: FastEthernet (10/100) LAN interface;
- ge: GigabitEthernet (10/100/1000) LAN interface;
- gr, gre: Generic routing encapsulation (GRE) interface for tunnel services—this interface is internally generated and not configurable;
- ip, ipip: IP-over-IP interface—this interface is internally generated and not configurable;
- lo: Loopback interface—this interface is internally generated and also configurable;
- ls, lsi: Link services interface—this interface is internally generated and not configurable;
- mt, mtun: Multicast GRE interface—this interface is internally generated and not configurable;
- pd, pimd: Protocol Independent Multicast (PIM) de-encapsulator interface—this interface is internally generated and not configurable;
- pe, pime: PIM encapsulator interface—this interface is internally generated and not configurable;

Continued on next page.

J-series Interface Naming (contd.)

- `pp`: Point-to-Point Protocol (PPP) interface—used for PPP over Ethernet (PPPoE);
- `se`: Serial interface (including EIA530, RS232, RS449, V.35, and X.21 interfaces);
- `sp`: Services interface;
- `tap`: This interface is internally generated and not configurable;
- `t1`: T1 (also called DS1) WAN interface; and
- `t3`: T3 (also called DS3) WAN interface.

J-series Interface Support

J2300 Router

Two FE ports and one of these fixed configurations:

- 2 x T1
- 2 x E1
- 2 x Serial

J4350/J6350 Routers

Four 10/100/1000 ports and up to six expansion PIMs:

- 2 x T1
- 2 x E1
- 2 x Serial
- 2 x FE
- 1 x DS3
- 1 x ADSL
- Many More

- Each PIM has its own IXP network processor
- Serial interface uses Juniper Networks proprietary cabling
 - Ten versions to choose from



Network Processor per PIM

Each PIM contains an Intel IXP 4xx network processor that handles some of the packet processing functions. Adding PIMs increases the overall capacity of the router.

Proprietary Serial Cabling

Serial interfaces use proprietary cables that allow auto detection of DTE/DCE and line protocol. The following cables are available:

- EIA530 DTE;
- EIA530 DCE;
- RS232 DTE;
- RS232 DCE;
- RS449 DTE;
- RS449 DCE;
- V.35 DTE;
- V.35 DCE;
- X.21 DTE; and
- X.21 DCE.

Selected Interface Features	
Interface Type	Features
Fast Ethernet	10/100 Mbps, full/half duplex, flow control, autonegotiation, auto-MDI/MDIX sensing, VLAN tagging, PPPoE, jumbo frames, and diagnostic capabilities
Serial	Autodetect DTE/DCE and line protocol based on cable (EIA530, RS232, RS449, V.35, and X.21), internal, external, and loop timing (8 MHz maximum), Frame Relay, PPP, and Cisco HDLC encapsulations, and diagnostic capabilities, independent clocking for every port on an interface
DS1/E1	D4 (SF) and ESF framing, fractional DS1/E1, AMI and B8ZS line coding, internal, external, and loop timing, Frame Relay, PPP, and Cisco HDLC encapsulations, line buildout, integral CSU, in-band diagnostics, alarm generation, integral BERT testing, and detailed statistics

Selected Interface Features

This slide details features supported by various J-series and M-series interfaces.

Note that some other vendors' products require all serial ports on the same line card to share the same clocking configuration; if different clocking requirements are needed, those products are unable to comply.

Agenda: Juniper Networks Enterprise Routers

- Customer Edge and Enterprise Platforms
- Overview of Enterprise Routing Platforms
- Architecture and Packet Flow
- Model Specifics
- Interface Support and Naming
- Field-Replaceable Units
- Network Management Options



Field-Replaceable Units

The slide highlights the topic we discuss next.

Field-Replaceable Units

- Refer to product documentation for step-by-step procedures
- Primary J-series FRUs:
 - Compact flash device
 - Power-off before installing or removing the primary compact flash drive
 - USB storage device
 - Ensure that the USB storage device is not in use before removal
 - DRAM
 - You must remove power before adding or removing DRAM
 - The J4350 and J6350 platforms support field-replaceable PIMs
 - PIM insertion or removal requires chassis power-off
 - The J6350 router supports hot-insertion and removal of redundant power supply
 - The J4350 and J6350 routers support an encryption acceleration card



Overview Only

This section simply provides an overview of the FRUs on J-series routers. For detailed replacement instructions, refer to the product documentation or Web-based training courses. Avoid safety hazards and equipment damage by always following proper procedures when replacing components.

Continued on next page.

Primary Field-Replaceable Units

J-series routers support a number of FRUs. These include the following:

- *Primary compact flash drive:* You can replace or upgrade the compact flash drive that provides primary storage on J-series platforms to a larger capacity. You must power-off the router before removing or installing the primary compact flash drive.
- *USB storage device:* You can remove or install the optional USB storage device while the router is operating. Before removal ensure that the router is not accessing the USB storage device.
- *DRAM:* You can replace or upgrade the dynamic random access memory on the RE to a larger capacity. Make sure you power-off and remove the power cord before taking the cover off the chassis.
- *PIMs:* PIMs are used only on the J4350 and J6350 chassis. They are not currently hot-swappable. You must power off the router before insertion or removal.
- *Power supply:* Only the J6350 power supplies are field-replaceable. On the J6350 router, power supplies are hot-swappable.
- *Crypto Acceleration Module:* The J4350 and J6350 routers support a hardware encryption acceleration card, which is field-replaceable. The router must be powered-off before insertion or removal.

FRU Examples



J2300 DRAM Installation



J2300 Primary CF Replacement



J6350 Power Supply Replacement



J6350 PIM Installation

FRU Examples

This slide illustrates some of the J-series FRUs. The M-series FRUs vary per router, and we discussed many of them earlier in this chapter in the detailed product descriptions. As always, see the product documentation or Web-based training for full procedures.

Agenda: Juniper Networks Enterprise Routers

- Customer Edge and Enterprise Platforms
- Overview of Enterprise Routing Platforms
- Architecture and Packet Flow
- Model Specifics
- Interface Support and Naming
- Field-Replaceable Units
- Network Management Options



Network Management Options

The slide highlights the topic we discuss next.

Network Management

- User interface options:
 - J-Web Web-based user interface
 - JUNOS software CLI
- Network management solutions:
 - SDX
 - JUNOScope
- SNMP
- RPM



User Interfaces

The traditional JUNOS software CLI gives access to all features. The J-Web user interface provides a graphical tool with quick configuration wizards for initial and common configuration tasks. The J-Web user interface is not intended to provide the full functionality found in the CLI.

Service Platforms

The Service Deployment System (SDX) software and JUNOScope are platforms that simplify the delivery of services across a network of J-series, M-series, and T-series routers.

SNMP

The JUNOS software can act as an SNMP agent. It supports SNMP versions 1, 2c, and 3. Several standard and Juniper Networks enterprise-specific management information bases (MIBs) are supported. See the Juniper Networks Web site for details about supported MIBs.

RPM

Real-time performance monitoring (RPM) is a tool that allows you and your customers to accurately measure the performance between two network endpoints. With the RPM feature, you configure and send probes to a specified target and monitor the analyzed results to determine packet loss, round-trip time, and jitter.

What Is XML?

- Extensible Markup Language is:
 - Derived from SGML
 - A toolkit for markup languages
 - A widely implemented standard
 - Go to <http://www.w3c.org/xml> for more information
- Documents written in XML are easily parsed
 - Overcomes problems with vendor-specific CLI grammars and syntax

```
<name>
  <first>Fred</first>
  <last>Flintstone</last>
</name>
```

The name of a certain Stone Age hero in unambiguous XML format



XML Definition

A markup language is a set of symbols that can be placed in the text of a document to demarcate and label the parts of that document. Markup languages organize information into a clear and unambiguous structure. In spite of its name, XML is not a markup language; rather, it's a toolkit for creating, shaping, and using markup languages.

Easily Parsed

A markup language created using XML rules is called an XML application. An XML application might describe proprietary information, but the way the information is described, using XML, is standardized. Because XML is an open standard that defines unambiguous structure, documents written in XML can be easily parsed for grammar and content by computer programs.

What Is JUNOScript?

- JUNOScript is an API that provides access to JUNOS software
 - JUNOScript is implemented as a proprietary XML application
- The JUNOS software CLI is actually a JUNOScript client
 - Therefore, what you can do using the CLI, you can also do using JUNOScript
 - The JUNOS software CLI outputs XML by piping to **display xml**
- Perl modules are available to simplify client development

```
lab@London> show configuration system host-name | display xml
<rpc-reply xmlns:junos="http://xml.juniper.net/junos/7.1R1/junos">
  <configuration>
    <system>
      <host-name>London</host-name>
    </system>
  </configuration>
  <cli>
    <banner></banner>
  </cli>
</rpc-reply>
```

A configuration snippet in XML format



JUNOScript API

JUNOScript provides a programmatic interface to JUNOS software. It is a proprietary markup language defined using XML rules.

JUNOScript Clients

All current J-series network management options (the CLI, J-Web, JUNOScope, and SDX software) are implemented as JUNOScript clients. They interface with the router using the XML-based JUNOScript application programming interface (API). In fact, the CLI can reveal the underlying XML by piping its output to **display xml**.

Available to Users

In addition, this API is documented and accessible by users who want to create their own software that interacts with and manages the router. Perl modules are available to simplify this software development.

What Is SDX Software?

- It is:
 - A service delivery platform
 - Service provisioning without SDX software is a manual process
 - SDX software provides a Web portal that can enable and customize services in (near) real time
 - A toolkit
 - Integrated applications including usage volume tracking and Web portals
 - Additional value-added services can be constructed by combining various SDX features
 - An enabler for new classes of business models
 - Allows providers to deliver new value-added products that were not previously possible

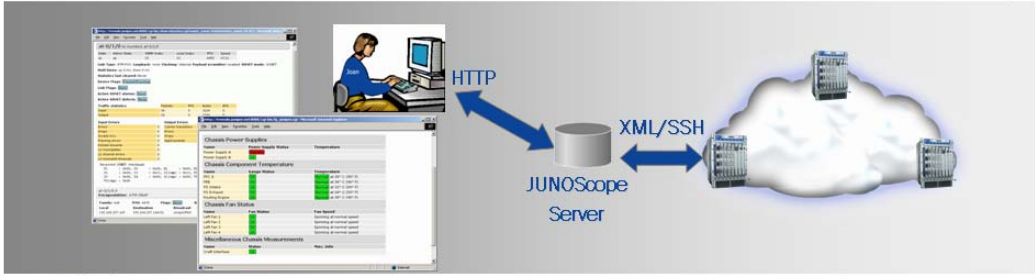


What Is SDX Software?

The Juniper Networks Service Deployment System (SDX) software is a robust, customizable product that allows a service provider's customers to dynamically activate value-added Internet services in real time. Consequently, service providers can instantly realize gains in revenue without significant effort from sales, operations, and provisioning teams. Using the SDX software, service providers can rapidly create and deploy many new value-added Internet services to hundreds of thousands of subscribers.

For more information on SDX software, go to <http://www.juniper.net/products/sdx/>, or attend the *Introduction to SDX-300 for JUNOS* training course.

What Is JUNOScope?



- It is:
 - A Web-based management application for J-series, M-series, and T-series routers
 - The foundation of the new J-Web interface
 - XML based
 - Features:
 - Router health monitoring suitable for NOC staff
 - Public domain tools and utilities available
 - Commercial configuration manager for router backups

Juniper your Net

Copyright © 2007 Juniper Networks, Inc. 2-52 Proprietary and Confidential www.juniper.net

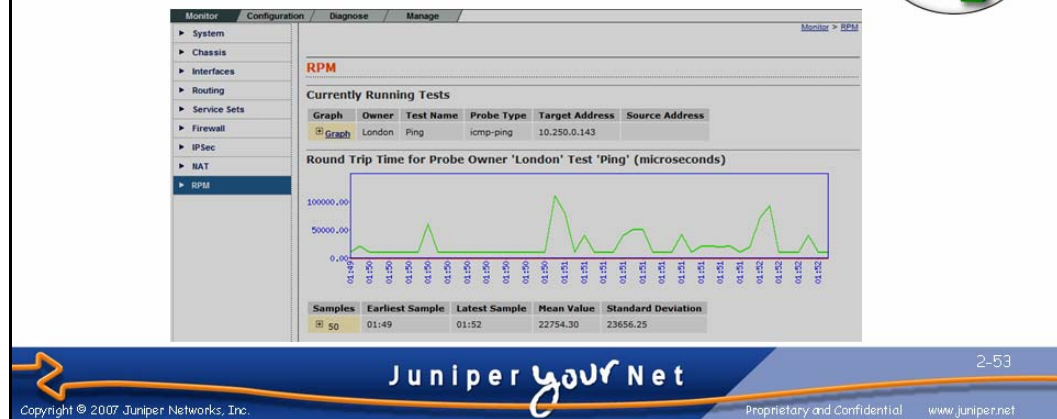
What IS JUNOScope?

JUNOScope is a management framework that consists of tools for managing IP services for the J-series, M-series, and T-series routing platforms. The integrated, out-of-the box tools can be deployed without any added development or customization. The set of tools provides support for multiple functions such as configuration management, inventory management, and system administration.

Users access JUNOScope using a Web-based interface, and users access JUNOScope interfaces with the JUNOS software on multiple J-series, M-series, and T-series routers using the JUNOScript API. The technology behind JUNOScope is the foundation for the J-Web interface. While JUNOScope allows configuration and management of a group of routers on a network, J-Web allows configuration and management of the local router only.

Real-Time Performance Monitoring

- Perform SLA monitoring
- Monitor RPM targets using:
 - HTTP GET commands
 - ICMP timestamp requests
 - ICMP, UDP, or TCP echo requests
- Graphical results using J-Web



SLA Monitoring

RPM is a tool for generating probe packets to monitor the network performance of a configured destination. You can use the results of these performance tests to confirm that service-level agreements (SLAs) are being met.

Supported Protocols

These probe packets can be sent using the following protocols:

- HTTP GET commands;
- ICMP timestamp request;
- ICMP echo request;
- UDP echo request; and
- TCP echo request.

The destination IP address must be configured to respond appropriately to the protocol used in the probe packet.

User Friendly Output

You can view the results of real-time performance monitoring graphically by navigating to the Monitor > RPM page in J-Web, as shown in the sample screen capture.

Review Questions

1. Describe applications that are well suited to J-series platforms.
2. What is the purpose of J-series copy protection, and how does it work?
3. Describe the architecture of J-series platforms.
4. Compare J-series platform offerings.
5. List interface options for J-series platforms.
6. Describe J-series interface naming.
7. List two network management options for J-series platforms, and briefly describe their capabilities.



This Chapter Discussed:

- Juniper Networks, Inc. enterprise products and their typical applications;
- General platform architecture;
- Juniper Networks router components;
- Packet flow;
- Interface support and naming conventions;
- Some FRUs; and
- Management options.



Operating Juniper Networks Routers in the Enterprise

Chapter 3: JUNOS User Interfaces

Chapter Objectives

- After completing this chapter, you will be able to:
 - Describe user interface options
 - Describe user authentication and authorization options
 - Differentiate active and candidate configurations
 - Use J-Web to configure and monitor a Juniper Networks router
 - Use the JUNOS software CLI to configure and monitor a Juniper Networks router



This Chapter Discusses:

- User interface options;
- User authentication and authorization;
- Active and candidate configurations;
- Using J-Web to configure and monitor a Juniper Networks router; and
- Using the CLI to configure and monitor a Juniper Networks router.

Agenda: JUNOS User Interfaces

- User Interface Options
 - User Authentication and Authorization
 - Active and Candidate Configurations
 - Using the J-Web Graphical User Interface
 - Using the JUNOS Software Command-Line Interface



User Interface Options

The slide lists the topics we cover in this chapter. We discuss the highlighted topic first.

User Interface Options

- **J-Web interface:**
 - A Web-based GUI
 - The J-Web service using HTTP is enabled by default on J-series routers
- **JUNOS software CLI:**
 - Available from console interface
 - RJ-45 RS-232 @ 9600 Bps, 8/1/N (not configurable)
 - Available by using the Telnet and SSH network protocols
 - Requires network interface and related service configuration
- **Dedicated Ethernet management port on M-series routers**
 - All J-series network ports support management access and transit traffic



J-Web Interface

J-Web is a Web-based graphical user interface (GUI) that you can access by either HTTP or HTTPS. It provides quick configuration wizards to simplify the most common configuration tasks. For more complicated configurations, the J-Web GUI allows you to directly edit the router's text configuration file. The J-Web GUI is installed and enabled by default on J-series routers. You can install the J-Web package on M-series routers.

JUNOS Software CLI

The JUNOS software CLI can be accessed over the network (in-band) by using the Telnet or SSH protocols. SSH versions 1 and 2 are supported, but 128-bit encryption is only available in the US export-controlled *domestic* JUNOS software images. JUNOS software CLI access is also available using an out-of-band serial console connection.

Continued on next page.

Dedicated Management Ethernet Port

All current M-series and T-series platforms have an Ethernet interface (fxp0) dedicated to network management. These routers do not forward traffic between the fxp0 management interface and any other network interface. This design ensures that no IP-level connectivity exists between the management network and the production network.

While we still recommend allocating a network interface for management purposes, J-series routers do not offer a dedicated port for this purpose. All J-series interfaces forward traffic by default. You can achieve similar functionality on the J-series platform by applying firewall (packet) filters to the interface allocated for network management.

Agenda: JUNOS User Interfaces

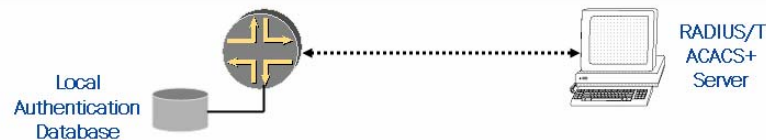
- User Interface Options
- User Authentication and Authorization
- Active and Candidate Configurations
- Using the J-Web Graphical User Interface
- Using the JUNOS Software Command-Line Interface



User Authentication and Authorization

The slide highlights the topic we cover next.

User Authentication



Local database

- Name and password
- Individual accounts and home directories

RADIUS and TACACS+

- Centralized authentication of users
- Users mapped to locally defined template users for authorization
- Extended regular expressions can be passed to alter authorization

Local

With local password authentication, you can individually configure usernames and passwords for each user to log in to the router. JUNOS software enforces the following password restrictions:

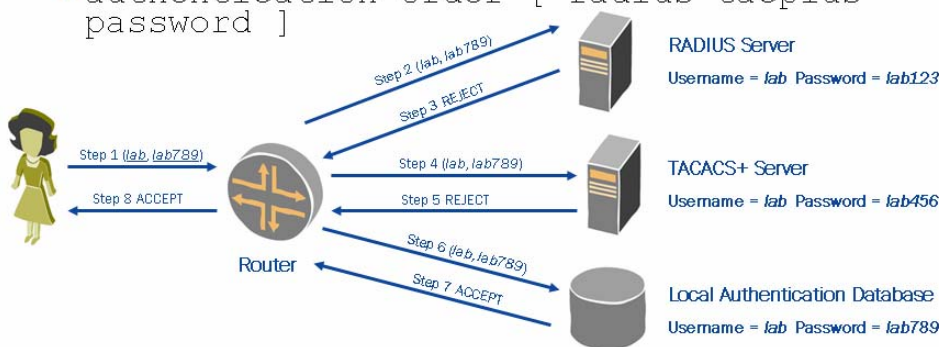
- The password must be at least 6 characters long.
- You can include most character classes in a password (alphabetic, numeric, and special characters), except control characters.
- Valid passwords must contain at least one change of case or character class.

RADIUS/TACACS+

RADIUS and TACACS+ are authentication methods used for validating users who attempt to access the router. They are both distributed client-server systems. The RADIUS and TACACS+ clients run on the Juniper Networks router; the server runs on a host connected to a remote network. Both protocols allow for user authentication. A locally defined user account determines authorization. Multiple RADIUS or TACACS+ authenticated users can be mapped to a locally defined user account. These local accounts are referred to as template users and avoid the need for each RADIUS or TACACS+ user to also have a locally defined user account. With the appropriate Juniper Networks extensions loaded on the server, both RADIUS and TACACS+ can override these template user authorization parameters by passing extended regular expressions to the router.

Authentication Order (1 of 3)

- Multiple authentication methods are supported
- Authentication order can be specified
 - The router tries each authentication method in order until the password is accepted
 - Even if a password is rejected, the router still tries the next configured authentication method!
 - If all configured authentication methods fail to reply, the router tries local authentication
- Example 1:
 - `authentication-order [radius tacplus password]`



Multiple Authentication Methods

You can configure the router to be both a RADIUS and TACACS+ client, and you can prioritize the order in which the software tries one or more of the three different authentication methods.

Authentication Order

For each login attempt, JUNOS software tries the authentication methods in order, until the password is accepted. The next method in the authentication order is consulted if the previous authentication method failed to reply or if the method rejected the login attempt. If no reply (accept or reject) is received from any of the listed authentication methods, JUNOS software consults local authentication as a last resort.

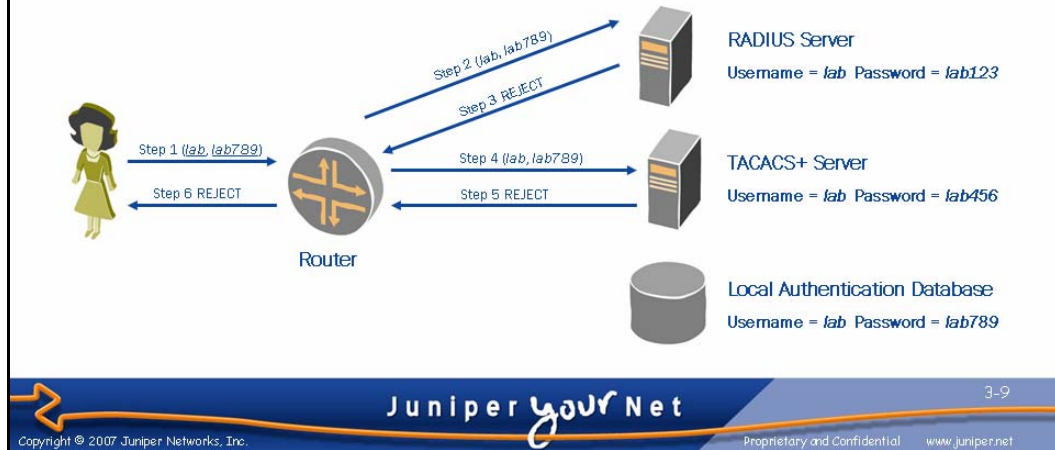
Example 1

In this example, we configured `authentication-order [radius tacplus password]`. We enter a username of `lab` and a password of `lab789`. We are successfully authenticated because each configured authentication method is attempted until the password is accepted by the local authentication database.

Authentication Order (2 of 3)

■ Example 2:

- `authentication-order [radius tacplus]`



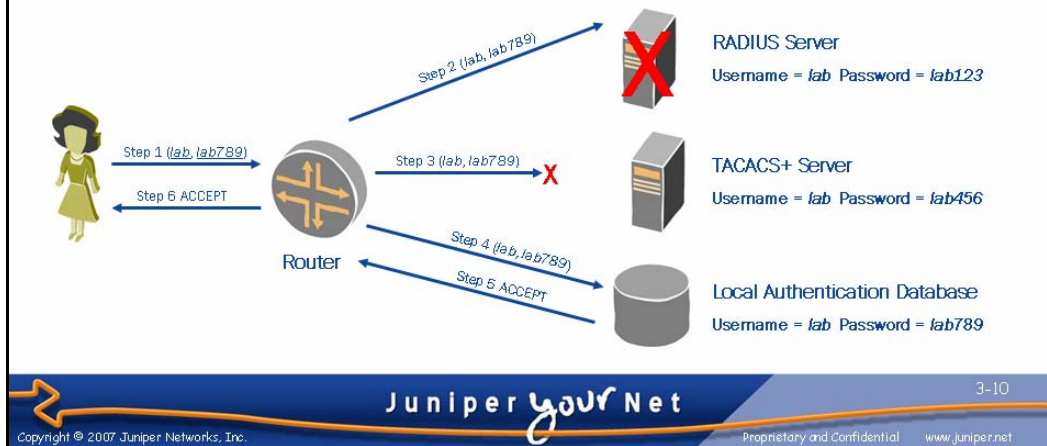
Example 2

In this example, we configured `authentication-order [radius tacplus]`. We enter a username of `lab` and a password of `lab789`. JUNOS software tries the password against the RADIUS server, which rejects it. It then tries it against the TACACS+ server, which also rejects it. JUNOS software does not consult local authentication because it is not listed in the authentication order, and at least one of the configured authentication methods did respond. The password is rejected.

Authentication Order (3 of 3)

■ Example 3:

- `authentication-order [radius tacplus]`



Example 3

In this example `authentication-order [radius tacplus]` is still configured. We enter a username of `lab` and a password of `lab789`. JUNOS software tries the password against the RADIUS server, which is down. The router receives no response, and after a timeout period, tries the TACACS+ server. A temporary network problem causes the TACACS+ server to be unreachable. After a timeout period, local authentication is consulted and the password is accepted. JUNOS software consults local authentication because none of the configured authentication methods responded.

Components of Authorization (1 of 2)



- Command and configuration statements are either authorized or denied
 - Applies to all nonroot users
 - Defined by a hierarchy of configuration components
- Users
 - Locally defined on the router
 - Member of a single class
- Class
 - A container for one or more permissions and explicit allow/deny overrides
 - Four predefined classes for common groups of permissions
 - Operator, read-only, super-user, unauthorized

Authorization Overview

Each command or configuration statement is subject to authorization. The router applies authorization to all nonroot users, and you cannot disable it. Authorization applies to both the J-Web and the command-line interfaces. Whether or not a command is authorized is defined by a configured hierarchy of authorization components as shown by the slide graphic.

Users

At the highest level, the configuration of user accounts on the router define authorization parameters. Multiple remotely authenticated users can be mapped to a locally defined template user. Users are members of a single login class.

Class

A login class is a named container that groups together a set of one or more permission flags. Login classes can also specify that the permission flags should be overridden for certain commands. Four predefined login classes exist to handle most situations. These classes and associated permission flags are the following:

- `super-user`: All permissions;
- `operator`: Clear, network, reset, trace, and view permissions;
- `read-only`: View permissions; and
- `unauthorized`: No permissions.

You can also create user-defined login classes for less common situations.

Components of Authorization (2 of 2)



■ Permissions

- Predefined sets of related commands

■ Allow and deny overrides

- Define exceptions for commands and configuration statements that would otherwise be allowed or denied
- Specified using regular expressions

Permissions

Several predefined permission flags group together the authorization of related commands. These predefined permissions and their definitions are the following:

- `access`: Allows viewing of network access configuration;
- `access-control`: Allows modifying of network access configuration;
- `admin`: Allows viewing of user accounts;
- `admin-control`: Allows modifying user accounts;
- `all`: Enables all permission bits to be turned on;
- `clear`: Allows clearing of learned network information;
- `configure`: Allows entering of configuration mode;
- `control`: Allows modifying of any configuration values;
- `field`: Is a special for field (debug) support;
- `firewall`: Allows viewing of firewall configuration;
- `firewall-control`: Allows modifying of firewall configuration;
- `floppy`: Allows reading and writing to the floppy drive;
- `interface`: Allows viewing of interface configuration;
- `interface-control`: Allows modifying of interface configuration;
- `maintenance`: Allows performing of system maintenance (as wheel);

Continued on next page.

Permissions (contd.)

- `network`: Allows network access;
- `reset`: Allows resetting and restarting of interfaces and processes;
- `rollback`: Allows ability to rollback for depth greater than zero;
- `routing`: Allows viewing of routing configuration;
- `routing-control`: Allows modifying of routing configuration;
- `secret`: Allows viewing of secret configuration;
- `secret-control`: Allows modifying of secret configuration;
- `security`: Allows viewing of security configuration;
- `security-control`: Allows modifying of security configuration;
- `shell`: Allows starting of a local shell;
- `snmp`: Allows viewing of SNMP configuration;
- `snmp-control`: Allows modifying of SNMP configuration;
- `system`: Allows viewing of system configuration;
- `system-control`: Allows modifying of system configuration;
- `trace`: Allows viewing of trace file settings;
- `trace-control`: Allows modifying of trace file settings;
- `view`: Allows viewing of current values and statistics; and
- `view-configuration`: Allows viewing of all configuration (not including secrets).

Allow and Deny Overrides

You can use the **deny-commands**, **allow-commands**, **deny-configuration**, and **allow-configuration** statements to define regular expressions that match operational commands or configuration statements. Matches are explicitly allowed or denied, regardless of whether the corresponding permission flags are set. You apply the **deny-** statements before the corresponding **allow-** statements, resulting in the authorization of commands that match both.

Authorization Example

```

graph LR
    user --> class
    class --> permissions
    permissions --> deny["deny-commands  
deny-configuration"]
    deny --> allow["allow-commands  
allow-configuration"]
    allow --> result["authorized  
or  
denied"]
    
```

```

root@host> show configuration system login
class noc {
    permissions view;
    allow-commands "clear interface statistics";
    deny-commands "clear interface statistics all";
}
user sue {
    uid 2000;
    class noc;
    authentication {
        encrypted-password
        "$1$UK4021d6$PZo./nQZbzIHxw7sYF/y3/";
    }
}
    
```

Juniper *your* Net

Copyright © 2007 Juniper Networks, Inc. Proprietary and Confidential www.juniper.net

3-14

Authorization Example

The configuration example on the slide shows how the various authorization components are configured:

- User *sue* is a member of the *noc* class.
- The *noc* class has *view* permissions.
- In addition, the *noc* class can clear statistics on individual interfaces using the **clear interface statistics interface-name** command.
- However, the *noc* class is denied the ability to clear the statistics of all interfaces at once with the **clear interface statistics all** command.

Agenda: JUNOS User Interfaces

- User Interface Options
- User Authentication and Authorization
- Active and Candidate Configurations
- Using the J-Web Graphical User Interface
- Using the JUNOS Software Command-Line Interface



Active and Candidate Configurations

This slide highlights the topic we discuss next.

Active and Candidate Configurations

- Batch configuration model:
 - Must commit configuration changes
- Active configuration:
 - Current operational configuration
 - Boot up configuration
- Candidate configuration:
 - A working copy for configuration changes
 - Initialized with the active configuration
 - Becomes active configuration upon commit



Batch Configuration Changes

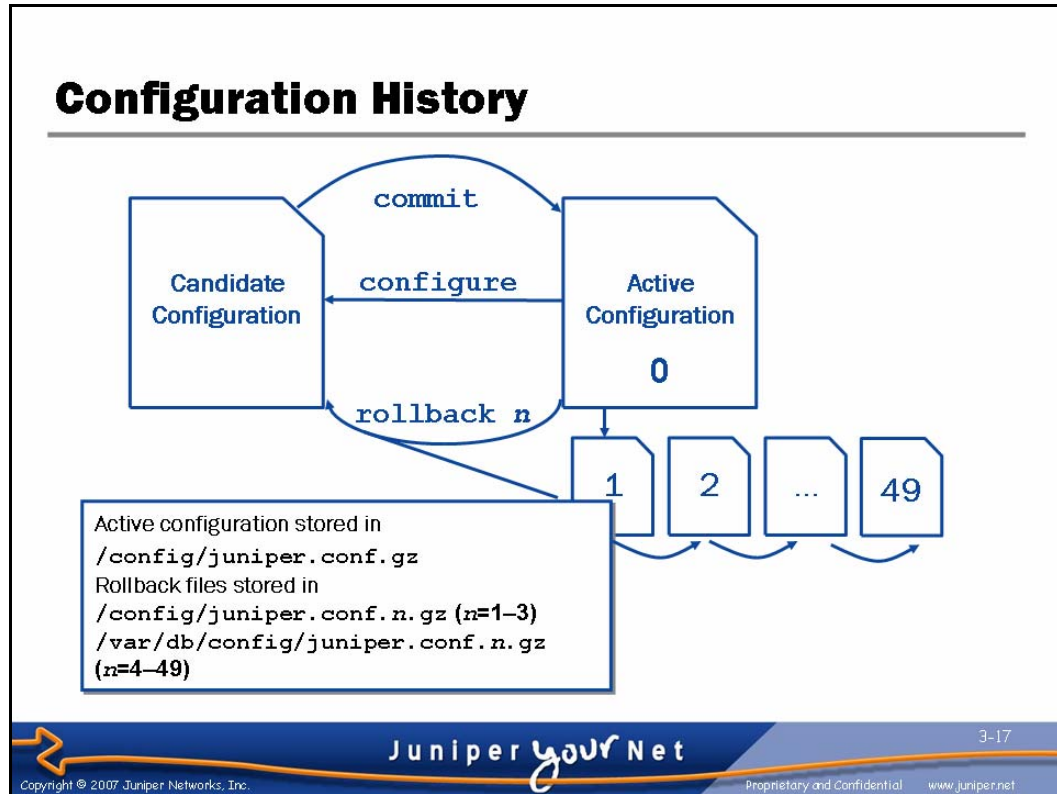
Unlike some router software, configuration changes to the JUNOS software do not take affect immediately. This design feature allows you to group together and apply multiple configuration changes to the running configuration as a single unit.

Active Configuration

The active configuration is the configuration currently operational on the router. It is also the configuration the router loads during the boot sequence. This concept is analogous to both the *running configuration* and *startup configuration* in other router software.

Candidate Configuration

The candidate configuration is a temporary configuration that might possibly become the active configuration. When you configure the router, a candidate configuration is created and initially populated with the router's active configuration. You then modify the candidate configuration. Once satisfied with your modifications, you can apply or commit the changes. This action causes the candidate configuration to become the active configuration.



Configuration Files and Configuration History

The **configure** command causes a *candidate* configuration to be created and populated with the contents of the *active* configuration. You can then modify the candidate configuration with your changes.

To have a candidate configuration take effect, you must commit the changes. At this time, JUNOS software checks the candidate configuration for proper syntax and it installs it as the *active* configuration. If the syntax is not correct, an error message indicates the location of the error, and no part of the configuration is activated. You must correct the errors before recommitting the configuration.

Changes you make to the candidate configuration are visible immediately. By default, there is only one candidate configuration. If multiple users are editing the configuration at the same time, all users can see all changes. If more than one user is modifying the configuration, committing it saves and activates the changes of all the users.

JUNOS software maintains a configuration history by storing previously active configurations. A maximum of 50 configurations are saved. This number includes the current *active* configuration, which is also known as **rollback 0**. You can easily recover previous configurations with a **rollback n** command.

Committing a configuration causes the old active configuration to become **rollback 1**. Each existing backup is renumbered and pushed further out, storing the oldest copy as number 49. The first three rollbacks (1-3) are stored in the `/config` directory, and the remainder are stored in the `/var/db/config` directory.

Agenda: JUNOS User Interfaces

- User Interface Options
- User Authentication and Authorization
- Active and Candidate Configurations
- ➔ Using the J-Web Graphical User Interface
- Using the JUNOS Software Command-Line Interface



Using the J-Web Graphical User Interface

The slide highlights the topic we discuss next.

The J-Web Graphical User Interface

- **Easy-to-use, Web-based graphical interface**
 - Operational monitoring, configuration, and routine maintenance actions
 - HTTP and HTTPS (SSL) support
 - No additional client software required
- **On J-series platforms, works *out of the box* with a factory-default configuration**
 - Can provide a temporary DHCP service to accommodate initial J-Web access
 - Automatically directs user to setup wizard when a factory-default configuration is detected
- **J-Web features:**
 - Same authentication and authorization as CLI
 - 15-minute session timeout
 - One browser window per J-Web session



The J-Web Graphical Interface

The J-Web interface provides quick access to common monitoring, configuration, and maintenance tasks on J-series routers. The quick configuration wizards simplify initial setup and common configuration tasks for users new to Juniper Networks products. The J-Web interface can be accessed through an HTTP- or an HTTPS-enabled Web browser. It does not require any additional software to be installed on the client workstation. An SSL certificate must be installed to enable HTTPS. Additionally, the domestic version of JUNOS software is needed to support 128-bit encryption; 56-bit encryption is supported in the export version of JUNOS software.

Enabled by Default

On J-series routers, the factory-default configuration enables autoinstallation and J-Web access via HTTP. If autoinstallation is unable to acquire an IP address and configuration, the router assigns itself an IP address and becomes a DHCP server on the built-in Fast Ethernet or Gigabit Ethernet interfaces. This design allows initial setup of the router to be easily accomplished from the Web browser of a directly attached workstation. When running a factory-default configuration, the router automatically directs J-Web users to the *Quick Configuration Setup* wizard.

Continued on next page.

J-Web Features

JUNOS software does not require separate authentication and authorization configurations for each user interface. A single configuration applies to both the CLI and J-Web interfaces. While CLI sessions, by default, can remain idle indefinitely, J-Web sessions are automatically timed out after 15 minutes of inactivity. Multiple, simultaneous J-Web sessions are supported, but opening multiple browser windows for a single session (by selecting *open link in new window*) can cause unpredictable results.

J-Web Capabilities

- **Quick Configuration wizards**
 - Initial setup, interfaces, routing, firewall/NAT, IPSec, etc.
- **Configuration maintenance**
 - History, compare, view, upload, download, full clickable edit
- **System monitoring**
 - System, chassis, interfaces, firewall, etc.
- **Fault isolation**
 - Ping and traceroute
- **System management**
 - Software upgrade, file system maintenance, license management, reboots and shutdowns



Wizards for Common Tasks

Quick Configuration wizards are provided for common configuration tasks and protocols. These wizards do not support advanced configurations. For example, you can configure only a single BGP peer using the routing wizard.

Configuration Maintenance

The Edit Configuration hierarchy allows more complicated configurations to be configured using the J-Web GUI. In addition, the J-Web interface offers tools to manage router configurations. You can view a history of previous configurations, compare two configurations, view the configuration in text format, and upload or download configuration files.

System Monitoring

You can also use the J-Web interface to monitor the health and operation of the router. Statistics are available for virtually every router component and operation.

Continued on next page.

Fault Isolation

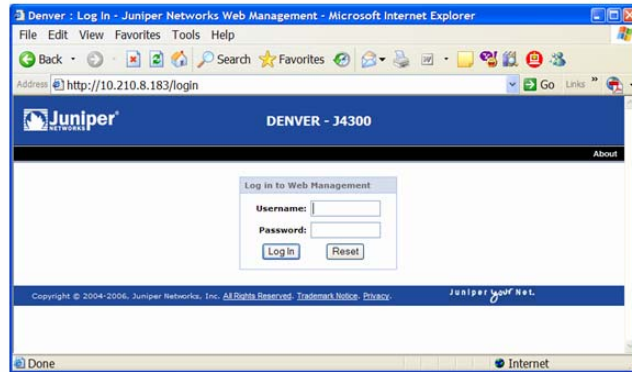
The GUI also provides access to the ping and traceroute utilities for diagnosing network problems.

System Management

You can also accomplish software upgrades, file system cleanup, license installation, and other common system management tasks from the J-Web interface.

J-Web Login

- J-Web sessions require a valid login
 - Uses the same authentication methods as CLI
 - Exception is initial access, when no login is needed to access the setup wizard

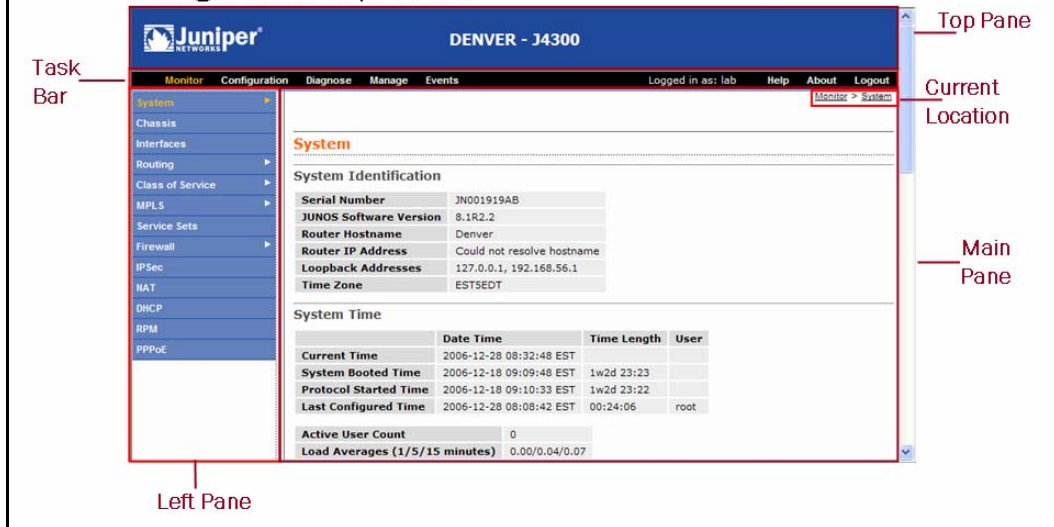


J-Web Login

After initial configuration the router requires all J-Web sessions to be authenticated. You can use the same login credentials as when accessing the router using the console, Telnet, or SSH. After entering your username and password click the Log In button to proceed.

J-Web Layout

- The J-Web page is composed of multiple panes
 - Monitor > System is the default view when a complete configuration is present



J-Web Window Layout

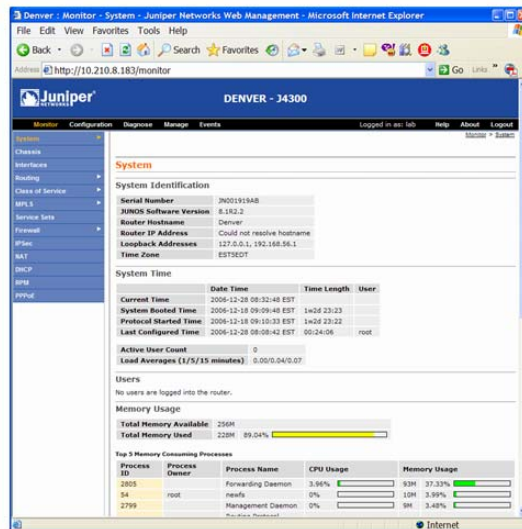
The J-Web browser window is organized into five panes to allow quick and simple navigation. These panes are the following:

- **Top pane:** Displays identifying information. This identifying information includes the Juniper Networks logo and the hostname and model of the router.
- **Task bar:** Contains tabs that identify the four top-level tasks of the J-Web hierarchy. The tab for the currently selected task is highlighted. Clicking a tab in the task bar navigates to the selected top-level task. Additionally, the right side of the task bar contains the current username and links that you can use to access the router's online help, J-Web version information, or log out of the J-Web GUI.
- **Left pane:** Displays suboptions of the currently selected task. The currently selected suboption is highlighted and its contents are displayed in the main pane. You can navigate between suboptions by clicking a suboption in the left pane.
- **Main pane:** Displays information about the currently selected suboption and allows you to enter information in text boxes, make selections, and click buttons.

When a factory-default configuration is present, the J-Web interface defaults to the Configuration > Quick Configuration > Setup hierarchy. Once initial setup is complete, it defaults to the Monitor > System hierarchy upon login. Your current location within the J-Web hierarchy is displayed in the top-right corner of the main pane.

The J-Web Monitor Task

- View the operation of the router and its protocols



Monitor

The J-Web Monitor task offers a view into the operation of the router and its protocols. Suboptions of the Monitor task include the following:

- System:** Provides information about memory, CPU, and storage usage. It also shows identifying information about the router, time information, and the user logged in through the CLI.
- Chassis:** Shows the hardware configuration and current operating status of router components.
- Interfaces:** Details interface configuration, current operational status, and performance statistics.
- Routing:** Views the routing table and routing protocol specific information.
- Class of Service:** Details (CoS) configuration.
- MPLS:** Details MPLS configuration.
- Service Sets:** Details configured service sets.
- Firewall:** Shows statistics on firewall rules and flows.
- IPSec:** Displays information about IPSec encryption and configured IPSec tunnels.

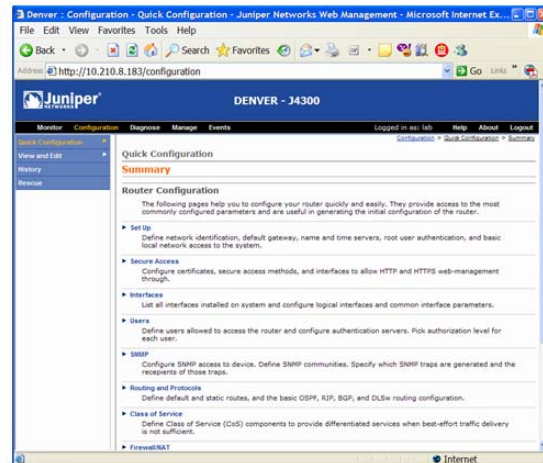
Continued on next page.

Monitor (contd.)

- NAT: Provides information about Network Address Translation (NAT).
- DHCP: Provides information about the configuration of the Dynamic Host Configuration Protocol (DHCP) server functionality.
- RPM: Displays real-time performance monitoring statistics and graphs for configured targets.
- PPPoE: Provides information and statistics about Point-to-Point Protocol over Ethernet (PPPoE) interfaces.

The J-Web Configuration Task

- Use *Quick Configuration* wizards
- Navigate a clickable view and edit function
- Access previous configuration history (rollbacks)
- Set a rescue configuration



Quick Configuration

The J-Web Configuration task offers multiple ways to view and modify the router's active configuration. Configuration > Quick Configuration wizards simplify the configuration of common features including initial setup, interfaces, users, routing, firewall/NAT, IPSec, and more.

View and Edit

The Configuration > View and Edit suboption allows you to work directly with the configuration file in the same text format used by the CLI, or you can use a clickable view that lets you drill down and configure any level of the configuration hierarchy.

History

The Configuration > History suboption allows you to view, download, or activate any of the 50 most recently committed configurations.

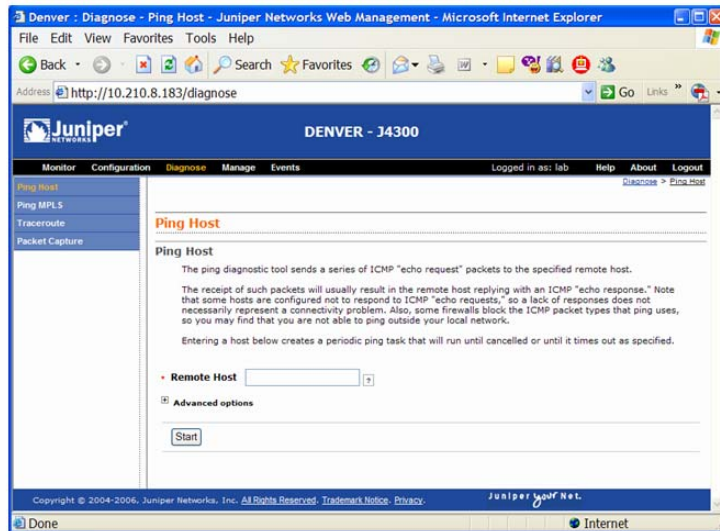
Continued on next page.

Rescue

The rescue configuration is a user-defined configuration that should ensure management access to the router. By default, there is no rescue configuration. Once a known-good configuration is activated on the router, you can use the Configuration > Rescue hierarchy to save the active configuration as the rescue configuration. Once a rescue configuration is set, pressing and immediately releasing the recessed CONFIG button on the front of a J-series router will load and commit the rescue configuration.

The J-Web Diagnose Task

- Access the ping, traceroute and packet capture utilities
 - Optional switches available through Advanced Options

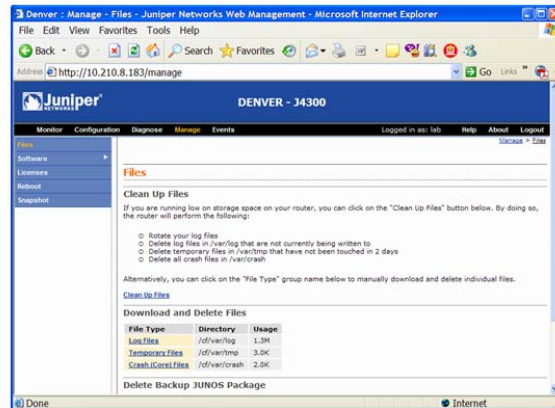


Troubleshooting

The J-Web Diagnose task offers a graphical interface to the ping, traceroute, and packet capture utilities. The ping and traceroute tools assist in troubleshooting network reachability problems, while you can use the packet capture utility to analyze traffic destined to a router (or transiting a J-series router). The Advanced options allow you to control various ping and traceroute parameters helpful in advanced troubleshooting.

The J-Web Manage Task

- Download and delete files
- Upgrade software
- Install and manage licenses
- Schedule system reboots
- Perform backups of software and configuration files



Files

The J-Web Manage > Files suboption allows files on the router's compact flash drive to be downloaded or deleted. A Clean Up Files wizard rotates logs and deletes unnecessary files.

Software

The Manage > Software suboption allows you to upgrade or downgrade the JUNOS software. You can upload JUNOS software from the local client or a remote FTP or HTTP server.

Licenses

The Manage > Licenses suboption allows you to view, add, and delete J-series licenses. It also provides a summary of licenses that are needed but not installed.

Reboot

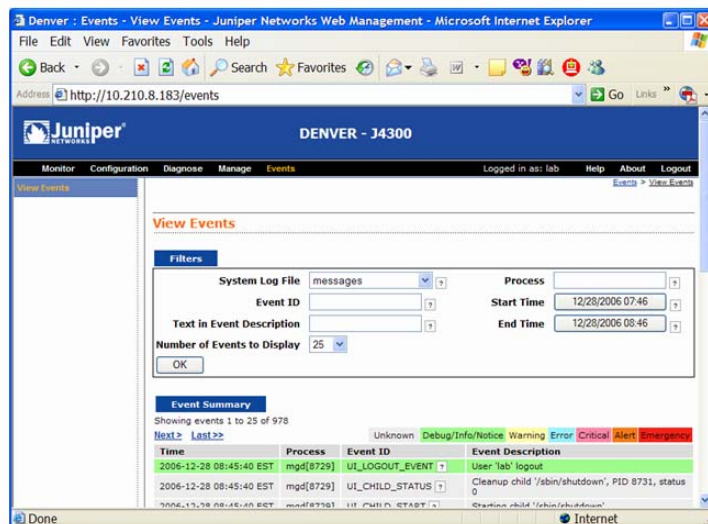
The Manage > Reboot suboption allows you to reboot or halt the router. These operations can be initiated immediately or scheduled for a future time.

Snapshot

The Manage > Snapshot suboption allows you to save the router's files to alternate media such as a USB drive.

The J-Web Events Task

- Provides access to log files



Events

The J-web Events tab provides access to view log files. You can view events from multiple log files and filter based on various criteria.

Initial Setup

- The Quick Configuration Set Up wizard makes initial configuration a snap

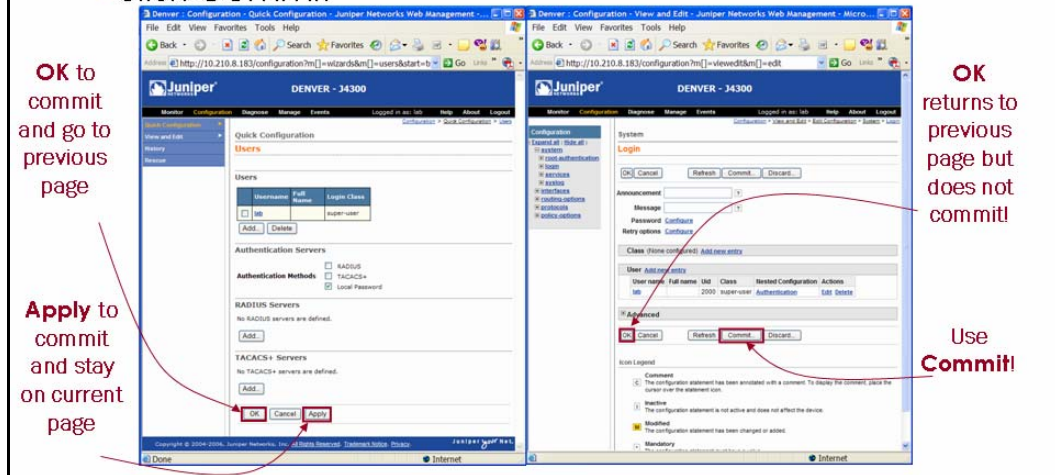
The screenshot shows the Juniper Quick Configuration Set Up wizard in a web browser. The browser title is "Denver : Configuration - Quick Configuration - Juniper Networks Web Management - Microsoft Internet Explo...". The address bar shows "http://10.210.8.183/configuration?m[]=wizard&m[]=setup&start=true". The Juniper logo is in the top left, and "DENVER - 34300" is in the top right. The interface has a sidebar with "Monitor", "Configuration", "Diagnose", "Manage", and "Events". The "Configuration" section is expanded, showing "Quick Configuration" and "Set Up". The main content area is divided into sections: "Identification" (Host Name: Denver, Domain Name, Root Password, Verify Root Password), "Time" (Time Zone: EST/EDT, NTP Servers, Current System Time: 12/28/2006 08:48), "Network" (DNS Name Servers, Domain Search, Default Gateway, Loopback Address: 192.168.56.1/32, fe-0/0/0.0 Address: 10.210.8.183/28), and "Management Access" (Allow Telnet Access, Allow JUNOScript over Clear-Text Access, Allow SSH Access). At the bottom are "OK", "Cancel", and "Apply" buttons.

Getting Started

The Configuration > Quick Configuration > Set Up wizard provides a single screen to input required and common configuration parameters. Fields marked with a red asterisk (*) are required. Once you have entered the desired values, click OK or Apply to commit your configuration.

J-Web Configuration Changes

- In *Quick Configuration* wizards:
 - Click OK or Apply to commit and activate configuration changes
- In the View and Edit hierarchy:
 - Click Commit



Activating Wizard Changes

When using Configuration > Quick Configuration wizards, clicking either OK or Apply commits and activates your changes. OK also returns you to the previous page, while Apply leaves you on the current page.

Activating View and Edit Changes

The OK button in the Configuration > View and Edit hierarchy behaves differently. It returns you to the previous page, but it does *not* commit your changes! Be sure to use the Commit button when making changes at the Configuration > View and Edit hierarchy.

Viewing Configuration History

- Use the J-Web Configuration > History page to view change log and to compare configuration files

Compare current configuration to previous (rollback 1)

Denver - Configuration - History - Juniper Networks Web Management - Microsoft Inter...

File Edit View Favorites Tools Help

Address http://10.210.8.183/configuration?m[]=history

Juniper DENVER - J4300

Monitor Configuration Diagnose Manage Events Logged in as: lab Help About Logout

Quick Configuration View and Edit History Reconfigure

History

Database Information
No users are editing the configuration database.

Configuration History
The following table shows the router's commit history.
To view a configuration, click the revision number.
To compare configurations, select two and click "Compare".

Compare

Number	Date/Time	User	Client	Comment	Log Message	Action
<input checked="" type="checkbox"/> Current	2006-12-28 08:28:42 EST	root	cli			Download
<input type="checkbox"/> 1	2006-12-28 08:07:04 EST	root	cli			Download Rollback
<input type="checkbox"/> 2	2006-12-28 07:32:17 EST	root	cli			Download Rollback
<input type="checkbox"/> 3	2006-12-28 06:54:14 EST	root	cli			Download Rollback
<input type="checkbox"/> 4	2006-12-28 05:38:28 EST	root	cli			Download Rollback

Internet

Viewing Configuration History

You can use the J-Web interface to show the differences between the current (active) configuration and the first rollback file.

Lab 1, Parts 1–3: The J-Web Interface

- Familiarize yourself with the J-Web user interface.



Lab 1, Parts 1–3: The J-Web Interface

The slide shows the objectives for this lab.

Agenda: JUNOS User Interfaces

- User Interface Options
- User Authentication and Authorization
- Active and Candidate Configurations
- Using the J-Web Graphical User Interface
- ➔ Using the JUNOS Software Command-Line Interface



Using the JUNOS Software Command-Line Interface

The highlights the topic we cover next.

CLI Modes and Feature Overview

- CLI operational mode:
 - Editing command lines
 - Command completion and history
 - Context-sensitive and documentation-based help
 - UNIX-style pipes
- CLI configuration mode:
 - Object-oriented hierarchy
 - Jumping between levels
 - Candidate configuration with sanity checking
 - Automatic rollback capability
 - Showing portions of configuration while configuring
 - Saving, loading, and deleting configuration files
 - Running operational-mode commands from within configuration



CLI Operational Mode

Use the CLI operational mode to monitor and troubleshoot the operation of the router.

CLI Configuration Mode

Use the CLI configuration mode when actually modifying the router's configuration.

CLI Modes

■ Operational mode:

- Monitor and troubleshoot the software, network connectivity, and router hardware

```
user@host>
```

The > character identifies operational mode

■ Configuration mode:

- Configure the router, including interfaces, general routing information, routing protocols, user access, and system hardware properties

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

The # character identifies configuration mode



Operational Mode

In operational mode, you use the CLI to monitor and troubleshoot the router. The **monitor**, **ping**, **show**, **test**, and **traceroute** commands let you display information and statistics about the software running on the router, such as routing table entries, and these commands let you test network connectivity.

Configuration Mode

You configure JUNOS software by entering configuration mode and creating a hierarchy of configuration statements. You can configure all properties of JUNOS software, including interfaces, general routing information, routing protocols, and user access, as well as several system hardware properties.

Logging In

- When logging in:

- Nonroot users are placed into CLI automatically

```
host (ttyd0)
```

```
login: user
```

```
Password:
```

```
--- JUNOS 7.1R2.2 built 2005-03-25 04:25:13 UTC
```

```
user@host>
```

- The root user must start CLI from shell
 - Do not forget to exit root shell after logging out of the CLI!

```
host (ttyd0)
```

```
login: root
```

```
Password:
```

```
--- JUNOS 7.1R2.2 built 2005-03-25 04:25:13 UTC
```

```
root@host% cli
```

```
root@host>
```

Shell Prompt

CLI Prompt



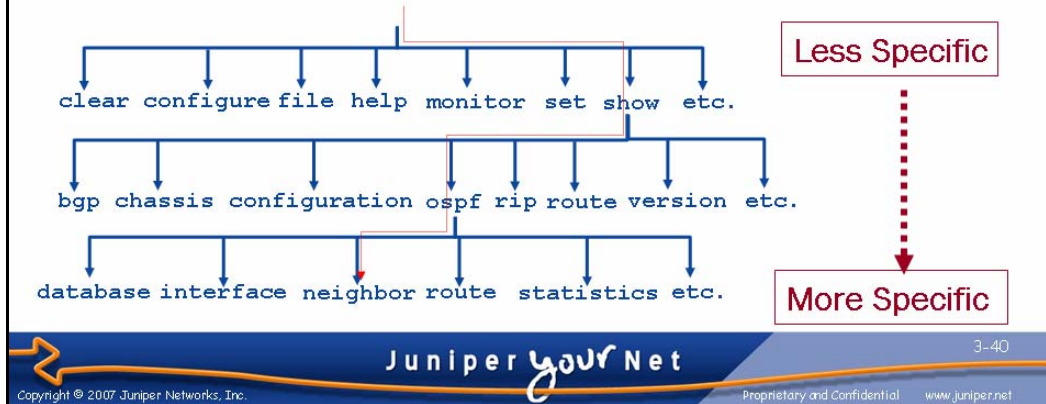
Logging In

JUNOS software requires a username and password for access. The router administrator creates user accounts and assigns permissions. J-series platforms have only the root user configured by default, without any password.

When you log in as the root user you are placed at the UNIX shell. You must start the CLI by typing the **cli** command. Then when you exit the CLI, you return to the UNIX shell. For security reasons, make sure you also log out of the shell using the **exit** command.

CLI Operational Mode

- Commands are executed (mainly) from the default CLI level (`user@host>`)
 - Can be executed from configuration mode with the **run** command
 - Hierarchy of commands
 - Example: **show ospf neighbor**



Operational Mode

You use operational-mode CLI commands to monitor and control the operation of the router. The operational-mode CLI commands are hierarchically structured, as shown on the slide. For example, the **show** command displays various types of information about the system and its environment. One of the possible options for the **show** command is **ospf**, which displays information about the OSPF routing protocol. Specifying the **neighbor** option, as in **show ospf neighbor**, outputs information on OSPF neighbors.

Continued on next page.

Operational Mode (contd.)

Key operational-mode capabilities include the following:

- Entering configuration mode;
- Controlling the CLI environment;
- Exiting the CLI;
- Monitoring and troubleshooting:
 - `clear`;
 - `monitor`;
 - `ping`;
 - `show`;
 - `test`; and
 - `traceroute`;
- Connecting to other network systems;
- Copying files;
- Restarting software processes; and
- Performing system-level operations.

Editing Command Lines

- EMACS-style editing sequences are supported

Keyboard sequence

→

```


user@host> show interfaces
user@host> show interfaces
user@host> show interfaces
user@host> show interfaces
user@host> show interfaces
        
```

Cursor position

▲

- Ctrl-b
- Ctrl-a
- Ctrl-f
- Ctrl-e

- The default VT100 terminal type also supports cursor positioning with the arrow keys



Juniper *your* Net

3-42

Copyright © 2007 Juniper Networks, Inc.

Proprietary and Confidential

www.juniper.net

EMACS-Style Control Keys

The CLI supports EMACS-style keyboard sequences that allow you to move around on a command line and delete specific characters or words. The following sequences are supported:

- Ctrl-b: Moves cursor left one character;
- Ctrl-a: Moves cursor to the beginning of the command line;
- Ctrl-f: Moves cursor right one character;
- Ctrl-e: Moves cursor to the end of the command line;
- Delete/Backspace: Deletes character before cursor;
- Ctrl-d: Deletes character over the cursor;
- Ctrl-k: Deletes from cursor to end of line;
- Ctrl-u: Deletes all characters/negates current command;
- Ctrl-w: Deletes entire word to left of cursor;
- Ctrl-l: Redraws the current line; and
- Ctrl-p/Ctrl-n: Repeats previous and next command in command history.

Continued on next page.

VT100 Terminal Type

JUNOS software defaults to a VT100 terminal type. This terminal type enables use of keyboard Arrow keys without any additional session or configuration modification.

Command and Variable Completion

■ Spacebar completes a command

```
user@host> sh<space>ow i<space>
'i' is ambiguous.
Possible completions:
  igmp      Show Internet Group Management Protocol...
  ike       Show Internet Key Exchange information
  interfaces Show interface information
  ipsec     Show IP Security information
  isis      Show Intermediate System-to-Intermediate...
```

Enter a space to complete a command

```
user@host> show i
```

■ Use the Tab key to complete an assigned variable

```
[edit policy-options]
user@host# show policy-statement t<tab>his-is-my-policy
then accept;

[edit policy-options]
user@host#
```

Use Tab to complete assigned variables



Space Completion for Commands

The CLI provides a completion function. Therefore, you do not always have to type the full command or command option name for the CLI to recognize it.

To complete a command or option that you have typed partially, press the Spacebar. If the partially typed letters begin a string that uniquely identifies a command, the CLI displays the complete command name. Otherwise, the CLI beeps to indicate that you have entered an ambiguous command, and it displays the possible completions.

The command completion option is on by default, but you can turn it off.

Tab Completion for Variables and Commands

You can also use the Tab key to complete variables. Examples of variables include policy names, AS paths, community names, and IP addresses.

Context-Sensitive Help

- Type a question mark (?) anywhere on command line

```
user@host> ?
Possible completions:
  clear          Clear information in the system
  configure      Manipulate software configuration
  information
  file           Perform file operations
  help           Provide help information
  . . .
user@host> clear ?
Possible completions:
  arp            Clear address resolution information
  bfd            Clear Bidirectional Forwarding
                 information
  bgp            Clear Border Gateway Protocol
  information
  firewall       Clear firewall counters
  . . .
```



Need Help?

The 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 Juniper Networks CLI, type a question mark (?). You do not need to press Enter. If you type the question mark at the command-line prompt, the CLI lists the available commands and options. If you type the question mark after entering the complete name of a command or an option, the CLI lists the available commands and options and then redisplay the command name and options that you typed. If you type the question mark in the middle of a command name, the CLI lists possible command completions that match the letters you have entered so far, then redisplay the letters that you typed.

Topical Help

- The **help topic** command provides information on general concepts

```
user@host> help topic interfaces ?
```

Possible completions:

accept-data	Accept packets destined for virtual IP...
accept-source-mac	Policers for specific source MAC addresses
access-profile	Mapping peer name and secrets for CHAP
accounting-profile	Accounting profile
acknowledge-timer	Maximum time to wait for link...
address	Interface address and destination prefix

...

```
user@host> 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, you configure the interface's IP address. For the iso family, you configure one or more addresses for the loopback interface. For the ccc, tcc, mpls, tnp, and vpls families, you never configure an address.

...



Help on General Concepts

There are various ways to use the **help** command. The **help topic** command displays usage guidelines for the statement. In the example on the slide, we are receiving information on configuring an interface address.

Help with Configuration Syntax

- Use **help reference** for assistance with configuration syntax

```
user@host> help reference interfaces address
address
```

Syntax

```
address address {
  arp ip-address (mac | multicast-mac) mac-address <publish>;
  broadcast address;
  destination address;
  destination-profile name;
  eui-64;
  multipoint-destination address dlci dlci-identifier;
  ...
}
```

Hierarchy Level

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

Description

```
Configure the interface address.
...
```



Help on JUNOS Software Configuration

The **help reference** command displays summary information for the statement. In other words, it contains JUNOS software-specific, configuration-related information. In the example on the slide, once again, we are using the **help** command for information on interface addressing. Notice the difference between the **help reference** command shown here and the **help topic** command from the previous slide.

Using | (Pipe)

- The pipe function allows you to filter and manipulate command output
 - Available in all modes and contexts

```
user@host> show route | ?
Possible completions:
  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
user@host> show route |
```



Using Pipe

For operational and configuration commands that display output, such as the **show** commands, you can filter the output. When help is displayed for these commands, one of the options listed is **|**, called a pipe, which allows the command output to be filtered. To filter the output of an operational-mode or a configuration-mode command, add a pipe and option to the end of the command. The options are the following:

- **compare (filename | rollback n)**: Available in configuration mode only using the **show** command. Compares configuration changes with another configuration file.
- **count**: Displays the number of lines in the output.
- **display detail**: Available in configuration mode only. Displays additional information about the contents of the configuration.
- **display xml**: Displays the output in JUNOScript XML format.
- **except regular-expression**: Ignores text matching a regular expression when searching the output. If the regular expression contains spaces, operators, or wildcard characters, you must enclose it in quotation marks.

Continued on next page.

The Pipe Commands (contd.)

- **find regular-expression**: Displays the output starting at the first occurrence of text matching a regular expression. If the regular expression contains spaces, operators, or wildcard characters, you must enclose it in quotation marks.
- **hold**: Holds text without exiting the --(more)-- prompt.
- **last**: Displays the last screen of information.
- **match regular-expression**: Searches for text matching a regular expression. If the regular expression contains spaces, operators, or wildcard characters, you must enclose it in quotation marks.
- **no-more**: Displays output all at once rather than one screen at a time.
- **request message**: Also sends the output to another users terminal.
- **resolve**: Converts IP addresses to DNS names. Truncates to fit original size unless you specify **full-names**.
- **save filename**: Saves the output to a file or URL.
- **trim**: Trims specified number of columns from the start line.

CLI Configuration Mode

- Where we are going...
 - Active vs. candidate configuration
 - Configuration history
 - Configuration mode
 - Navigating configuration hierarchy
 - Making or deleting configuration changes
 - Viewing configuration differences
 - Saving and loading configuration files



CLI Configuration Mode

The slide shows the topics examined on the following pages.

Review: Active Versus Candidate Configuration

- Batch configuration model:
 - Must commit configuration changes
- Active configuration:
 - Current operational configuration
 - Boot up configuration
- Candidate configuration:
 - A working copy for configuration changes
 - Initialized with the active configuration
 - Becomes active configuration upon commit



Batch Configuration Changes

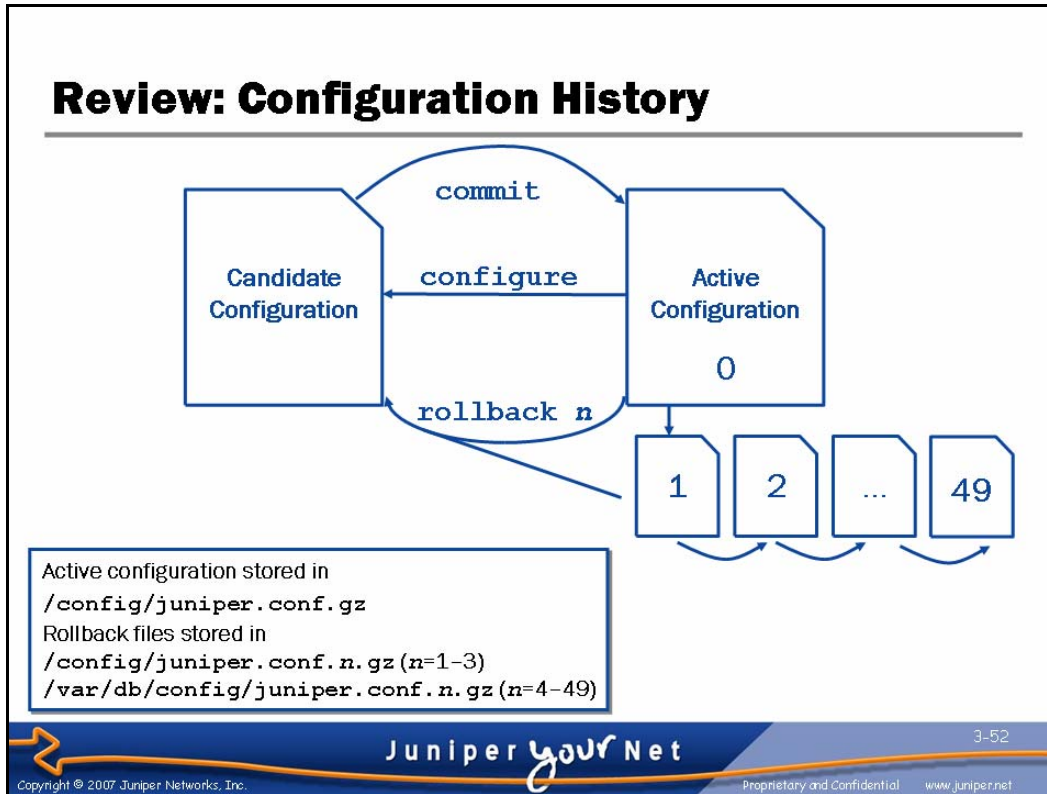
Unlike some router software, configuration changes to the JUNOS software do not take affect immediately. This design feature allows you to group together and apply multiple configuration changes to the running configuration as a single unit.

Active Configuration

The active configuration is the configuration currently operational on the router. It is also the configuration the router loads during the boot sequence. This concept is analogous to both the *running configuration* and *startup configuration* in other router software.

Candidate Configuration

The candidate configuration is a temporary configuration that might possibly become the active configuration. When you configure the router a candidate configuration is created and initially populated with the router's active configuration. You then modify the candidate configuration. Once satisfied with your modifications, you can apply or commit the changes. This action causes the candidate configuration to become the active configuration.



Configuration Files and Configuration History

The **configure** command causes a *candidate* configuration to be created and populated with the contents of the *active* configuration. You can then modify the candidate configuration with your changes.

To have a candidate configuration take effect, you must commit the changes. At this time, JUNOS software checks the candidate configuration for proper syntax and it installs it as the *active* configuration. If the syntax is not correct, an error message indicates the location of the error, and no part of the configuration is activated. You must correct the errors before recommitting the configuration.

Changes you make to the candidate configuration are visible immediately. By default, there is only one candidate configuration. If multiple users are editing the configuration at the same time, all users can see all changes. If more than one user is modifying the configuration, committing it saves and activates the changes of all the users.

JUNOS software maintains a configuration history by storing previously active configurations. A maximum of 50 configurations are saved. This number includes the current *active* configuration, which is also known as **rollback 0**. You can easily recover previous configurations recovered with a **rollback n** command.

Committing a configuration causes the old active configuration to become **rollback 1**. Each existing backup is renumbered and pushed further out, storing the oldest copy as number 49. The first three rollbacks (1-3) are stored in the `/config` directory, and the remainder are stored in the `/var/db/config` directory.

Entering Configuration Mode

- Type **configure** or **edit** at the CLI operational-mode prompt:

```
user@host> configure
Entering configuration mode
```

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

- To allow a single user to edit the configuration, type **configure exclusive**
- **configure private** allows the user to edit a private copy of the candidate configuration
 - Multiple users can edit private candidate configurations simultaneously
 - At commit time, the user's private changes are merged back into the global configuration



Starting Configuration Mode

You enter configuration mode by issuing the **configure** command or the **edit** command from the CLI's operational mode. If, when you enter configuration mode, another user is also in configuration mode, a message indicates who the user is and what portion of the configuration the user is viewing or editing.

In configuration mode, the prompt changes from the angle bracket (>) of operational mode to the octothorp (#), preceded by the name of the user and the name of the router.

The portion of the prompt in brackets, such as [edit], is a banner indicating that you are in configuration mode and specifying your location within the statement hierarchy.

Exclusive Configuration

By default, multiple users can enter configuration mode and commit changes. To allow only a single user to edit the configuration, use the **configure exclusive** command.

Continued on next page.

Private Configuration

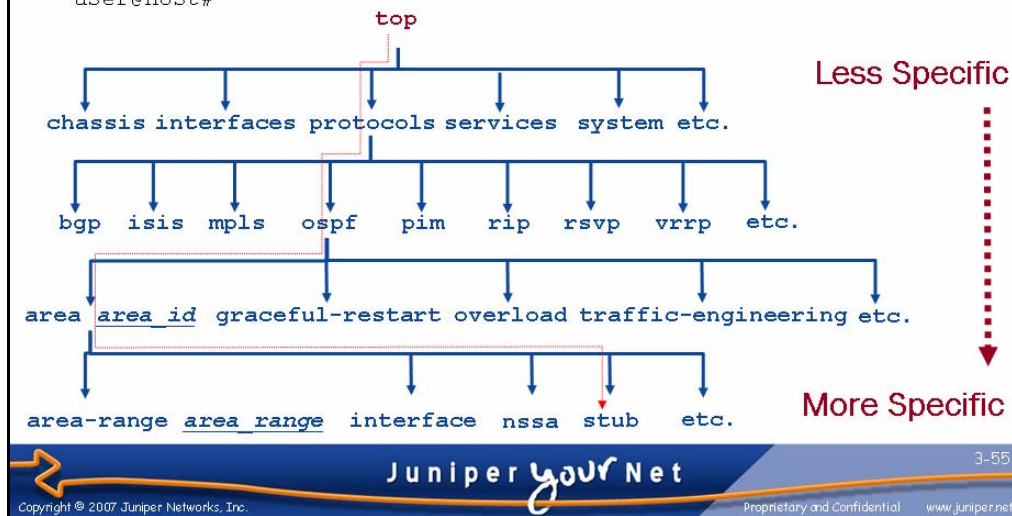
Entering configuration mode using **configure private** allows multiple users to edit the configuration while only committing their private changes (you must issue a **commit** command from the [edit] hierarchy). If private users issue a **rollback 0** command, only their changes are discarded. If two users are in private mode and both make the same change (*user_1* changes the system hostname to *foo* while *user_2* sets the name to *bar*), the second **commit** will fail with an error message to avoid configuration conflicts. The second user's changes are placed into effect if a second **commit** is issued, however.

When a user is in private mode, other users must enter private mode or use **configure exclusive** to become the master, or they cannot modify the candidate configuration. Exiting private configuration without committing changes results in the loss of any modifications made to the private candidate configuration.

Configuration Statement Hierarchy

```
[edit]
user@host# edit protocols ospf area 51 stub

[edit protocols ospf area 0.0.0.51 stub]
user@host#
```



Statement Hierarchy

In configuration mode, you enter commands that affect the statement hierarchy. The statement hierarchy stores configuration information and is independent of the CLI operational-mode command hierarchy. The commands available in configuration mode are also independent of the commands available in operational mode. For example, CLI operational mode includes a **show** command to display specific information, while CLI configuration mode provides a **show** command to display the statement hierarchy. The two commands are independent of each other.

The statement hierarchy is organized in a tree structure similar to Windows folders or UNIX directories, grouping related information into a particular branch of the tree.

Configuration File Is Hierarchical

- CLI commands are entered without curly braces

```
[edit system]
user@host# set services web-management http port 8080
```

- The result is a hierarchical configuration file, complete with curly braces

```
[edit system]
user@host# show services
web-management {
    http {
        port 8080;
    }
}

[edit system]
user@host#
```



Hierarchical Configuration

You use the **set** command in CLI configuration mode to modify the candidate configuration. The **show** command is used to display the candidate configuration. Both commands are relative to the current configuration hierarchy, shown by the `[edit]` prompt.

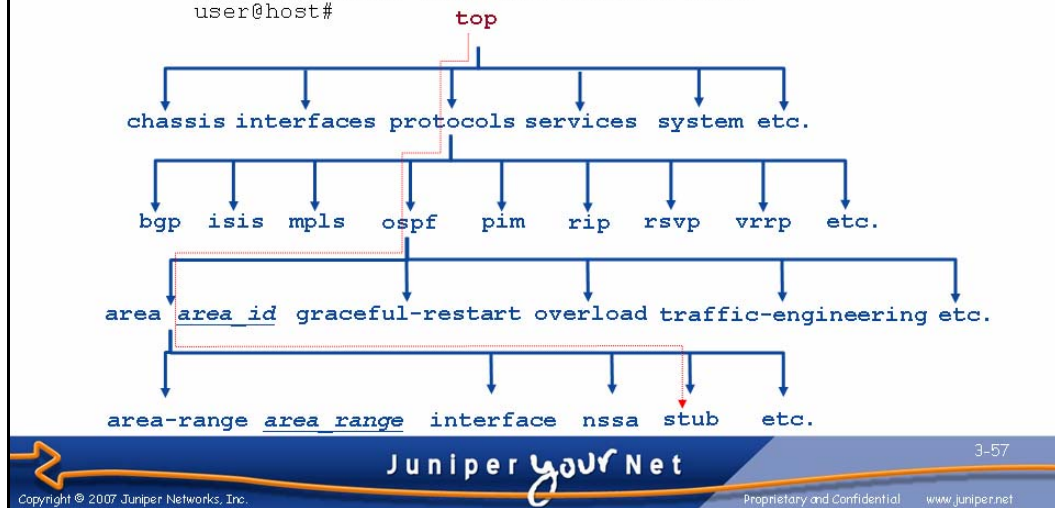
Configuration files use curly braces and indentation to visually display the hierarchical structure of the configuration. Terminating, or leaf, statements in the configuration hierarchy are displayed with a trailing semicolon. Neither the curly braces nor semicolons are entered in the **set** command.

Moving Between Levels (1 of 6)

- `edit` functions like a change directory (CD) command

```
[edit]
user@host# edit protocols ospf area 51 stub

[edit protocols ospf area 0.0.0.51 stub]
user@host#
```



Moving Between Levels Is Like Changing Directories

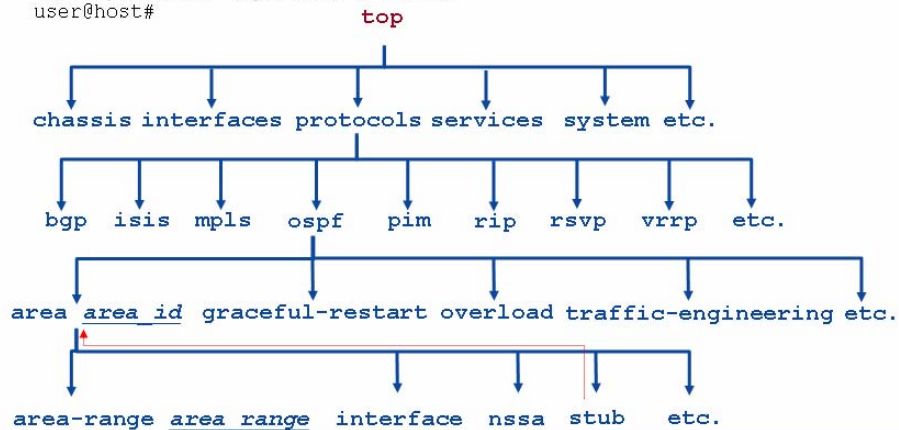
To move down through an existing configuration statement hierarchy or to create a hierarchy and move down to that level, use the **edit** command, specifying your desired hierarchy level. After you issue an **edit** command, the configuration mode banner changes to indicate your current level in the hierarchy.

Moving Between Levels (2 of 6)

- **up** moves up one level in the hierarchy

```
[edit protocols ospf area 0.0.0.51 stub]
user@host# up
```

```
[edit protocols ospf area 0.0.0.51]
user@host#
```



Moving Up One Level

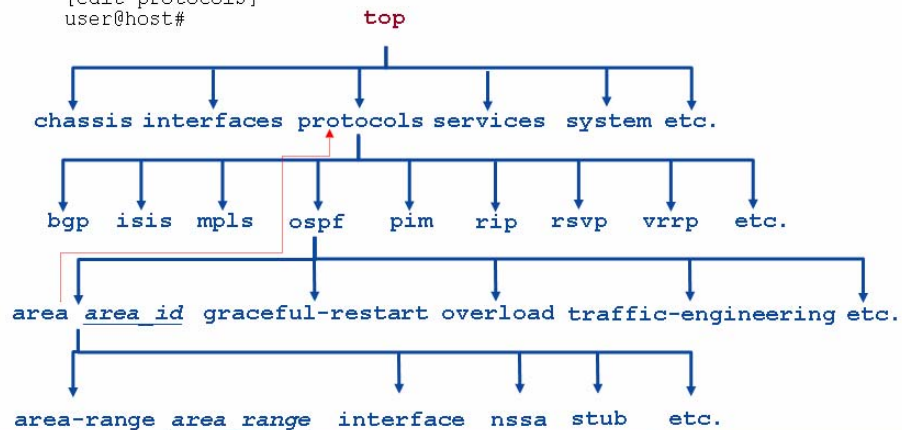
To move up one level from the current position in the hierarchy, use the **up** command.

Moving Between Levels (3 of 6)

- `up n` moves up n levels

```
[edit protocols ospf area 0.0.0.51]
user@host# up 2
```

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



Moving Up More Than One Level

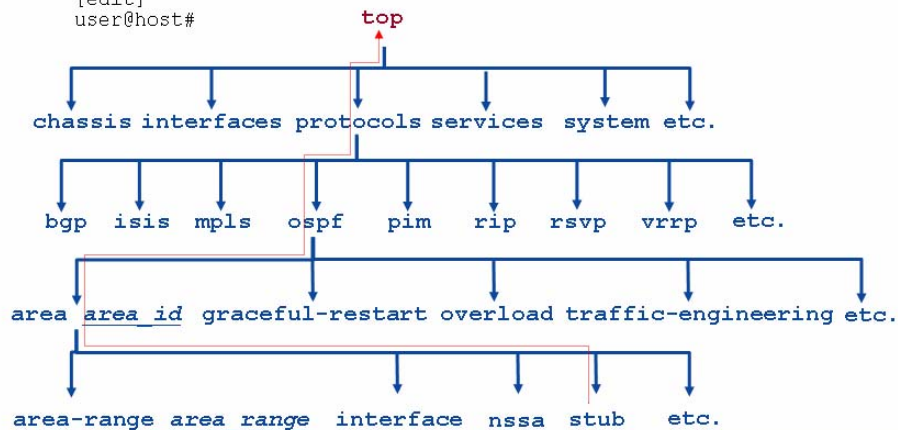
To move up more than one level from the current position in the hierarchy, supply an optional count to the `up` command. You will be moved up the number of levels specified or to the top of the hierarchy if there are fewer levels than specified.

Moving Between Levels (4 of 6)

- **top** moves to the top of the hierarchy

```
[edit protocols ospf area 0.0.0.51 stub]
user@host# top
```

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



Take Me to the Top

The **top** command quickly moves you to the top of the configuration hierarchy. **top** can be combined with **edit** to quickly move to a different hierarchy or with **show** to display a different hierarchy:

```
[edit protocols ospf area 0.0.0.51 stub]
user@host# top edit system login
```

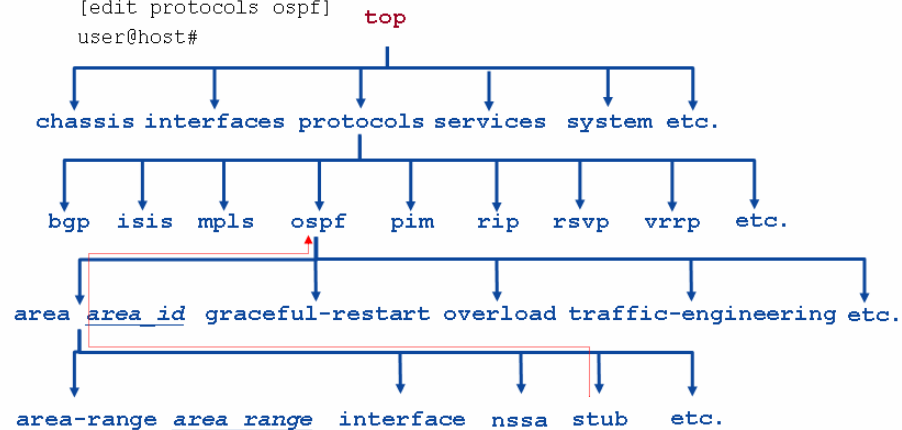
```
[edit system login]
user@host#
```

```
[edit protocols ospf area 0.0.0.51 stub]
user@host# top show system services
web-management {
  http {
    port 8080;
  }
}
```


Moving Between Levels (5 of 6)

- **exit** moves to the *previous higher* level in the hierarchy

```
[edit protocols ospf]
user@host# edit area 51 stub
[edit protocols ospf area 0.0.0.51 stub]
user@host# exit
[edit protocols ospf]
user@host#
```



Back to Where I Was Before

The **exit** command moves to the most recent higher level of the hierarchy. Entering **exit** at the top level of the hierarchy exits configuration mode. You can exit configuration mode from any level of the hierarchy by supplying the **configuration-mode** argument to the **exit** command:

```
[edit]
user@host# exit
Exiting configuration mode
```

```
[edit protocols ospf area 0.0.0.51 stub]
user@host# exit configuration-mode
Exiting configuration mode
```

```
user@host>
```

Moving Between Levels (6 of 6)

- Summary of moving between levels:
 - **edit** functions like a change directory (CD) command
 - **up** moves up one level
 - **up n** moves up n levels
 - **top** moves to the top of the hierarchy
 - **exit** moves to the *previous higher* level in the hierarchy or exits configuration mode if at the top level of the hierarchy

```
[edit]
user@host# edit protocols ospf area 51 stub
[edit protocols ospf area 0.0.0.51 stub]
user@host# up
[edit protocols ospf area 0.0.0.51]
user@host# up 2
[edit protocols]
user@host# top
[edit]
user@host# exit
The configuration has been changed but not committed
Exit with uncommitted changes? [yes,no] (yes)
```



In Summary

The **edit**, **up**, **top**, and **exit** commands let you quickly navigate between levels of the configuration hierarchy.

Viewing Candidate Configuration

```
[edit]
user@host# show system services
ssh;
web-management {
  http {
    port 8080;
  }
}
```

You can display just the portions that concern you from the root of the hierarchy...

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

```
[edit system services]
user@host# show
ssh;
web-management {
  http {
    port 8080;
  }
}
```

...or use **edit** to park yourself at a specific sub-hierarchy



Displaying the Candidate Configuration

To display the candidate configuration, use the configuration-mode **show** command. This command displays the configuration at the current hierarchy level or at the specified level below the current location.

The **show** command has the following syntax: **show statement-path**. When displaying the configuration, the CLI indents each subordinate hierarchy level, inserts curly braces to indicate the beginning and end of each hierarchy level, and places a semicolon (;) at the end of statements that are at the lowest level of the hierarchy. The display format is the same format you use when creating an ASCII configuration file, and it is also the same format that the CLI uses when saving a configuration to an ASCII file.

In cases where an empty statement leads to an invalid configuration because it is incomplete or meaningless, the **show** command does not display any of the statement path.

Identifying Configuration File Differences (1 of 2)

- Change the candidate configuration:

```
[edit system]
user@host# set services telnet
[edit system]
user@host# delete services web-management
[edit system]
user@host# delete services ssh
```

- Display differences between the candidate and active configurations:

```
user@host# show | compare
[edit system services]
-  ssh;
+  telnet;
-  web-management {
-    http {
-      port 8080;
-    }
-  }
```



Modifying a Candidate Configuration

The example on the slide modifies a candidate configuration by enabling Telnet access and removing SSH and J-Web access. **set** and **delete** commands are relative to the current hierarchy.

Viewing Differences

Piping the output of a **show** command to the CLI compare function displays the differences between the candidate configuration file and the active configuration, also known as `rollback 0`. Configuration comparison is *patch*-like. Thus, instead of showing the entire configuration and where changes were made, only the actual changes are shown. By using the pipe switch you can save the configuration differences to the file name of your choosing. Once saved, you can issue a **load patch filename** command to merge the contents of the patch file into the candidate configuration where they can be viewed, edited, and ultimately committed.

Identifying Configuration File Differences (2 of 2)

- Compare active and historical configurations

```
user@host> show configuration | compare rollback number
```

```
user@host> show configuration | compare filename
```

- Compare arbitrary files:

```
user@host> file compare files filename_1 filename_2
```



Comparing Active and Rollback Configurations

Using the operational-mode **show configuration | compare rollback number** command, as shown on the slide, allows you to view differences between the active configuration and any of the 49 rollback configurations. Similarly, the **show configuration | compare filename** command allows you to compare the active configuration to an arbitrary file. You can also use **show | compare rollback number** and **show | compare filename** in configuration mode to compare the *candidate* configuration with rollback configurations and arbitrary files respectively.

Viewing Differences in Other Files

The operational-mode **file compare files** command allows you to view differences between any two text files, including log files. The output of this command is in the same patch-like format as the **show configuration | compare** command.

Removing Statements (1 of 2)

- Statements added with **set** are removed with the **delete** command
 - Removes everything from the specified hierarchy down
 - Use **wildcard delete** to save time

```
user@host# show services
ssh;
web-management {
  http {
    port 8080;
  }
}
```

Note that the entire Web-management hierarchy is removed by the delete statement

```
[edit system]
user@host# delete services web-management
```

```
[edit system]
user@host# show services
ssh;
```



Removing Configuration Statements

Use the configuration-mode **delete** command to remove statements that were added to the configuration with a **set** command. This command deletes the statement and all its subordinate statements and identifiers. Deleting a statement or an identifier effectively *unconfigures* the functionality associated with that statement or identifier, returning that functionality to its default condition.

Consider using the **wildcard delete** function when deleting individual statements is too arduous and deleting an entire configuration sub-hierarchy lacks the granularity that is needed. Sample syntax for a **wildcard delete** is shown:

```
[edit]
user@host# wildcard delete interfaces fe-*
matched: fe-0/0/2
Delete 1 objects? [yes,no] (no) yes
```

In addition to deleting configuration statements, you should also consider the use of **deactivate** to cause the specified portion of the configuration hierarchy to be ignored, while still retaining the original configuration. Issue an **activate** command to place the configuration back into effect. Also consider the use of **disable** for interfaces. Use the **set** command to add a disable statement to flag a given interface as being administratively disabled.

Removing Statements (2 of 2)

- Pop quiz: You have just disabled an interface with a `set interface interface-name disable` statement. How do you re-enable this interface?



Pop Quiz!

Issue a `delete interface interface-name disable` command to delete the disable statement placed into effect with a `set` command. This syntax has been known to strike some folks as being a more than a bit on the double-negative side; then again, these same folks tend to agree that a `no shutdown` statement, as used for similar functionality on other vendors' equipment, is equally counter-intuitive!

Committing a Configuration (1 of 2)

- Configuration changes must be committed to take effect

```
[edit]
user@host# commit
commit complete
[edit]
user@host#
```

- Use **commit check** to confirm syntax

```
[edit]
user@host# commit check
[edit interfaces lo0 unit 0 family inet]
  'address 192.168.69.1/24'
    Loopback addresses' prefix must be 32 bits
error: configuration check-out failed
```

- Use **commit confirmed** to temporarily activate

```
user@host# commit confirmed
commit confirmed will be automatically rolled back in 10 minutes
unless confirmed
commit complete
```



Don't Forget to Commit

Remember, the router does not automatically apply your configuration changes. You must use the **commit** command to activate your candidate configuration.

Checking Configuration Syntax

When you commit a candidate configuration (which you can do from any hierarchy level), you commit the entire configuration in its current form. Use the **commit check** command to validate the syntax of a candidate configuration without actually placing it into effect.

Remote Configuration Is Risky

Of course, **commit check** cannot catch logical errors in your configuration. What happens when you are configuring a router remotely and make a mistake that leaves the router inaccessible to remote connections? This scenario is solved by the **commit confirmed** command. When you issue a **commit confirmed time-out** command, the system starts a timer, during which time it expects to see another commit. If a second **commit** does not occur within the time-out value specified (a range of 1 to 65,535 minutes is supported, with 10 minutes being the default), the system performs a **rollback 1, commit** sequence on your behalf. After the automatic rollback you can load the **rollback 1** file to look for your mistake.

Committing a Configuration (2 of 2)

- Schedule a future commit with `commit at`

```
[edit]
user@host# commit at 21:00:00
configuration check succeeds
commit at will be executed at 2005-05-25 21:00:02 UTC
Exiting configuration mode
```

- Add comments with `commit comment`

```
user@host# commit comment "Changed OSPF configuration"
commit complete
```

```
user@host> show system commit
0    2005-05-25 04:10:17 UTC by lab via cli
    Changed OSPF configuration
```

...

- Use `commit and-quit` to save time

```
[edit]
user@host# commit and-quit
commit complete
Exiting configuration mode
user@host>
```



Scheduled Commits

You can also schedule a commit that occurs at a specific time using the `commit at time` command. To view any pending commits (and the commit history) use the `show system commit` command. You can cancel a pending commit with the `clear system commit` command.

Adding a Log Entry to Your Commit

You can also add a log entry to your commit using the `commit comment "comment-string"` option. These logs are visible in the output of the `show system commit` command.

Exiting Configuration Mode

The `and-quit` option can be specified to the `commit` command to activate your changes and exit configuration mode in a single step.

Backing Out of Configuration Changes

- Use the **rollback** command to restore one of the last 50 previously committed configurations


```
[edit]
user@host# rollback
load complete
```
- Use **rollback** (or **rollback 0**) to reset the candidate configuration to the currently active configuration (which is the last version committed)
 - **rollback 1** loads the configuration before that
 - **rollback n** loads *n* configurations before that
- Using **rollback** only modifies the candidate configuration
 - Don't forget to commit the changes!



Backing Out of Changes

The software saves the last 50 committed versions of the configuration. To overwrite the candidate configuration with one of these previously committed versions, use the CLI configuration **rollback** command. By default, the system returns to the most recently committed configuration.

Specifying Rollback Files

To return to a version prior to the configuration most recently committed, include the version number in the **rollback** command:

```
[edit]
user@host# rollback version
load complete
[edit]
user@host#
```

The **version** argument can be a number in the range 0 through 49. The most recently saved configuration is version 0, which is a copy of the current active configuration. The oldest committed configuration that is now automatically saved is now version 49.

Continued on next page.

You Must Commit

The **rollback** command only modifies the candidate configuration. To activate the changes that you loaded, issue the **commit** command:

```
[edit]
```

```
user@host# commit
```

Saving Configuration Files

- Save current candidate configuration using the **save** command

```
[edit]
user@host# save filename
```

- File saved to user's home directory unless full path name is specified
- Only saves from the current hierarchy down
- File name can specify:
 - A path and filename on the local router's file system
 - A URL (FTP and SCP)
- Miscellaneous features:
 - **terminal** option for **save** commands
 - Simplifies load operations from terminal buffers
 - Pipe option for **display set**
 - Displays the **set** statements used to create a configuration
 - Periodic saves to a remote host



Saving Files

You can save the candidate configuration from your current configuration session to an ASCII file. Doing this saves the configuration in its current form, including any uncommitted changes. If more than one user is modifying the configuration, saving it saves the changes made by all the users.

Note that only configuration statements at the current hierarchy level and below are saved. To save the entire candidate configuration, you must be at the top level of the configuration hierarchy. By default, the CLI saves the configuration to the specified file in your home directory. For example, user *doug* would store files in the */var/home/doug* directory. You can change this default by specifying a path name.

Specifying File Names

You can specify a filename in one of the following ways:

- filename or path/filename.
- ftp://user:password@host/path/filename: Puts file in location explicitly described by this URL using the FTP protocol. Substituting the word prompt for the password causes the router to prompt you for the user's password.
- scp://user@host/path/filename: Puts file on a remote system using the SSH protocol. You will be prompted for user's password.

Continued on next page.

Miscellaneous Features

JUNOS software supports saving configuration data to a terminal device. With this option the appropriate configuration hierarchy name, curly brackets, and **replace** tag are added to readily accommodate pasting into another router's configuration using some form of load-terminal operation. You can also save the output to a file for later use in a file load operation. An example of **load terminal** at work is provided here:

```
[edit]
user@host# load replace terminal
[Type ^D at a new line to end input]
protocols {
replace: ospf {
    area 0.0.0.0 {
        interface fe-0/0/0.0;
        interface fe-0/0/1.0;
        interface se-0/0/2.0;
    }
}
}
load complete
```

Piping output to **display set** is supported. This feature converts a configuration into the actual **set** statements used to create the configuration; this option is intended to simplify the editing of configuration data being cut and pasted between routers:

```
[edit protocols ospf]
user@host# # show | display set
set protocols ospf area 0.0.0.0 interface fe-0/0/0.0
set protocols ospf area 0.0.0.0 interface fe-0/0/1.0
set protocols ospf area 0.0.0.0 interface se-0/0/2.0
```

You can configure either a periodic or commit-driven upload of the router's configuration to a particular host using FTP. A typical configuration is shown:

```
[edit system archival]
user@host# show
configuration {
    transfer-on-commit;
    archive-sites {
        "ftp://lab:lab@10.250.0.254";
    }
}
```

Note that because a destination file name is not specified in the FTP URL, the file written to the archive host takes the form of routername_juniper.conf_date_time.

Loading Configuration Files

- Configuration information can come from an ASCII file or terminal emulation capture buffer
- The **load** command supports various arguments:
 - Override an existing configuration:
 - **load override filename**
 - Merge new statements into current configuration:
 - **load merge filename**
 - Replace existing statements in current configuration:
 - **load replace filename**
 - Take input from terminal capture buffer:
 - **load (replace | merge | override) terminal**
 - Load relative to current configuration hierarchy:
 - **load (replace | merge) (filename | terminal) relative**
- Changes candidate configuration only
 - You must issue a **commit** to activate



Loading a Configuration

You can use the configuration-mode **load** command to load a complete or partial configuration from a local file, from a file on a remote machine, or from a terminal emulation program's capture buffer. The **load** command supports several arguments that determine the specifics of the operation.

Continued on next page.

Load Options

The following list provides details about the arguments to the **load** command:

- **merge**: Combines the current configuration with the configuration being loaded.
- **override**: Completely overwrites the current configuration with the configuration being loaded. You must perform override operations at the root of the configuration hierarchy.
- **replace**: Looks for a replace tag in the configuration being loaded. Existing statements of the same name are replaced with the those in the loaded configuration for stanzas marked with the **replace** tag.
- **terminal**: Uses the text you type at the terminal as input to the configuration. Type **Ctrl-d** to end terminal input. Usually this option is used in conjunction with a terminal emulation program's copy/paste functionality to copy and paste configuration data from one system to another.
- **relative**: Normally, a **load merge** or **load replace** operation requires that the data being loaded contain a full path to the related configuration hierarchy. The **relative** option negates this need by telling the router to assume that the data being loaded should be added *relative* to the current configuration hierarchy.

Changes Candidate Configuration Only

In all cases, after the **load** operation is complete, you must issue a **commit** to activate the changes made to the configuration.

run Is Cool

- Use the **run** command to execute operational-mode CLI commands from within configuration
 - Can be a real time-saver when testing the effect of a recent change

```
[edit interfaces fe-0/0/0]
lab@HongKong# set unit 0 family inet address 10.250.0.141/16
```

```
[edit interfaces fe-0/0/0]
lab@HongKong# commit
commit complete
```

Test configuration changes without
leaving configuration mode with **run**

```
[edit interfaces fe-0/0/0]
lab@HongKong# run ping 10.250.0.149 count 1
PING 10.250.0.149 (10.250.0.149): 56 data bytes
64 bytes from 10.250.0.149: icmp_seq=0 ttl=255 time=0.967 ms
```

```
--- 10.250.0.149 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.967/0.967/0.967/0.000 ms
```



Running with the Big Dogs (in the tall grass)

The **run** command allows you to execute operational-mode commands while in configuration mode. It is similar to the **do** command on other vendors' equipment. This extremely handy time-saver works for all operational-mode commands and is supported at all configuration hierarchies. In the example on the slide, the operator is editing the configuration for the router's `fe-0/0/0` interface. After assigning what is hoped to be the correct IP address, the operator commits the change (without the **confirmed** switch), and invokes the **run** command to execute a quick ping test.

Review Questions

1. Describe user authentication and authorization.
2. List two user interface options supported by J-series platforms.
3. Briefly describe the use of each main menu option supported by the J-Web application.
4. What does the `confirmed` switch do when committing changes?
5. What command restores the router to the previously committed configuration?
6. How can you display differences between an active and a candidate configuration?
7. What is the difference between the `merge`, `override`, and `replace` arguments when loading a file?
8. How can you display the status of an interface while in configuration mode?

This Chapter Discussed:

- User interface options;
- User authentication and authorization;
- Active and candidate configurations;
- Using J-Web to configure and monitor a Juniper Networks router; and
- Using the CLI to configure and monitor a Juniper Networks router.

Lab 1, Parts 4–5: The JUNOS Software CLI

- Familiarize yourself with the JUNOS software CLI.



Lab 1, Parts 4–5: The JUNOS Software CLI

The slide shows the objectives for this lab.



Operating Juniper Networks Routers in the Enterprise

Chapter 4: Installation and Initial Configuration

Chapter Objectives

- After successfully completing this chapter, you will be able to:
 - Install Juniper Networks routers
 - Use autoinstallation to load a configuration file from a server
 - Save and restore a rescue configuration
 - Return to a factory-default configuration
 - Perform initial configuration using J-Web
 - Perform initial configuration using the CLI



This Chapter Discusses:

- The general process and guidelines for installing Juniper Networks routers;
- Loading a configuration file, and saving and restoring rescue configurations;
- Returning the router to its factory-default configuration; and
- Performing initial configuration on the router using both J-Web and the CLI.

Agenda: Installation and Initial Configuration

- Installation Guidelines
 - Autoinstallation
 - Rescue and Factory-Default Configurations
 - Configuration Checklist
 - Initial Configuration Using J-Web
 - Initial Configuration Using the CLI
 - Overview of Interface Configuration
 - Configuring Interfaces Using J-Web



Installation Guidelines

The slide lists the topics we cover in this chapter. We discuss the highlighted topic first.

General Installation Guidelines

- Follow safety guidelines
- J2300 platform:
 - Desk, wall, or rack mountable
 - Maximum weight is 12 lbs (5.4 Kg)
 - Rack mounting brackets and rubber feet are included
- J4350, J6350, M7i, M10i, and M120 platforms:
 - Rack mountable only
 - Size and weight vary with platform
- Attach network and console cables
- Attach power cables



Safety Guidelines

Be sure to read and follow applicable safety guidelines before installing a J-series router. You can find these guidelines in the included *Quick Start* guide or online at <http://www.juniper.net/techpubs/>.

J2300 Router Installation

The J2300 Services Router can be mounted on a desk, on a wall, or in a rack. The included rubber feet are used for desk and wall mounting. The included mounting brackets are used for wall and rack mounting.

J4350, J6350, M7i, M10i, and M120 Router Installation

The remaining Juniper Networks enterprise routers are only rack mountable. The size and weight of the routers varies per device.

Continued on next page.

Connecting Cables

You can connect to the console using the provided console cable. Use a standard RJ-45 Ethernet cable, no crossover necessary, for connecting to the fixed Fast Ethernet interfaces. Remember that serial interfaces use proprietary Juniper Networks cables.

Attaching Power

AC-powered routers include an appropriate AC power cord for your geographic location.

Power On and Power Off

Recessed POWER
Button



- JUNOS software is a multitasking environment
 - A graceful shutdown of the OS ensures file system integrity
 - Use J-Web Manage > Reboot page or the **request system halt** CLI command to gracefully halt the operating system
 - Power is maintained to the system; reboot with console activity
- Power off from the front panel on J-series router
 - Briefly depress the power button for graceful OS shutdown and OS-triggered soft power off
 - Push and hold the power button for 4–5 seconds to remove power; the OS should be shut down first
- Power up with momentary push of the power button
 - Automatic power on after power is lost and restored

Gracefully Shut Down the JUNOS Software

The JUNOS software is a multitasking environment. To ensure file system integrity you should always gracefully shut down the router. Although unlikely, failure to gracefully shut down the router could possibly leave it unable to boot.

J-series POWER Button Operation

Briefly pressing the POWER button on the front panel will power on a router that is powered off. The POWER ON LED will light steadily green.

If the router is operating, briefly press the POWER button to initiate a graceful OS shutdown and power off. The green POWER ON LED will blink during the shutdown process and turn off once the shutdown is complete.

Holding the POWER button for 4 to 5 seconds immediately powers off the router without shutting down the JUNOS software. This process should only be done after first gracefully shutting down the OS from J-Web or the CLI.

The POWER button is a standby power switch. If AC power is attached, the router remains in standby mode and a small amount of standby voltage is still present in the chassis. Make sure you disconnect the AC power cord before opening the chassis.

Continued on next page.

Automatic Power On

If AC power to an operating router is interrupted, the router automatically powers on upon power restoration. The router does not require you to press the POWER button in this situation.

Agenda: Installation and Initial Configuration

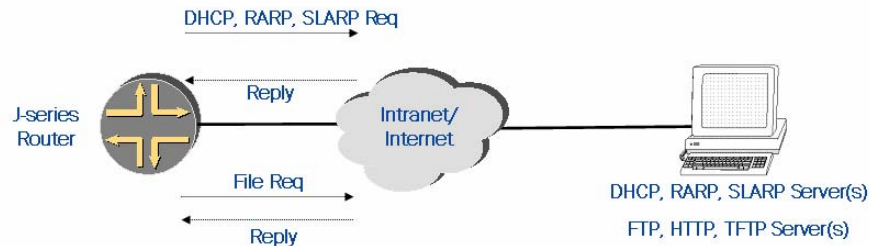
- Installation Guidelines
- Autoinstallation
- Rescue and Factory-Default Configurations
- Configuration Checklist
- Initial Configuration Using J-Web
- Initial Configuration Using the CLI
- Overview of Interface Configuration
- Configuring Interfaces Using J-Web



Autoinstallation

This slide highlights the topic we discuss next.

Autoinstallation



- Autoinstallation simplifies and automates initial router configuration (J-series platforms only)
 - Address acquisition
 - Configuration file retrieval
- Compatible with other vendor's AutoInstall

Autoinstallation Process

J-series routers include autoinstallation functionality. The primary purpose of autoinstallation is to install a configuration file on the router. This process facilitates the centralized management of router installation. The JUNOS software's `autod` process handles this task in two stages. The first stage acquires IP addresses for each interface. The second stage then uses those IP addresses to transfer and activate a configuration file.

Compatibility

Autoinstallation functionality is compatible with another vendor's AutoInstall feature.

When Is Autoinstallation Attempted?

- Autoinstallation is active only under certain circumstances:
 - Factory-default configuration
 - Partial (bootstrap) configuration and autoinstallation enabled

```
[edit]
user@host# load factory-default
warning: activating factory configuration

[edit]
user@host# show system
autoinstallation {
    delete-upon-commit; ## Deletes [system autoinstallation] upon change/commit
    traceoptions {
        level verbose;
        flag {
            all;
        }
    }
}
...

```



When Is Autoinstallation Active?

The autoinstallation process, `autod`, is not active by default. JUNOS software runs this process any time the `autoinstallation` stanza is configured under the `[edit system]` hierarchy. This stanza exists when the factory-default installation is active or when a partial (bootstrap) configuration that includes the `autoinstallation` stanza has been manually loaded on the router.

You can use the `show system autoinstallation status` command to check if autoinstallation is running:

```
user@host> show system autoinstallation status
warning: autoinstallation subsystem not running - not needed by configuration.

user@host>

```

Configuring Autoinstallation

- Bootstrap parameters override the default behavior
 - Use the **interfaces** keyword to specify which interfaces can be used for autoinstallation
 - Use the **configuration-servers** keyword to specify a list of URLs that are used to retrieve the configuration file

```
[edit system autoinstallation]
user@host# show
interfaces {
  fe-0/0/0 {
    rarp;
  }
}
configuration-servers {
  tftp://tftpserver.example.com/config.conf;
}
```

Annotations:

- Enables autoinstallation (points to `[edit system autoinstallation]`)
- Limits autoinstallation to the specified interface(s) (points to `fe-0/0/0`)
- Defines dynamic address acquisition protocol for a given interface (points to `rarp;`)
- URL list for configuration file retrieval (points to `tftp://tftpserver.example.com/config.conf;`)



Modifying Autoinstallation Behavior

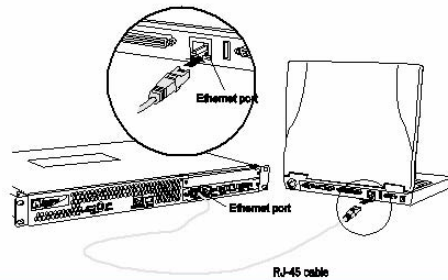
Parameters configured at the `[edit system autoinstallation]` hierarchy will override the default autoinstallation behavior. This design allows users to define a bootstrap configuration that allows autoinstallation to function in their specific environment.

Specifically, the **interfaces** keyword restricts which interfaces can be used for autoinstallation and potentially defines which address acquisition protocols can be used on those interfaces. By default, autoinstallation is attempted on all interfaces.

The **configuration-servers** keyword allows specific protocols, configuration servers, and configuration file names to be specified in the form of configuration server URLs.

Autoinstallation Address Acquisition

- Send out DHCP and RARP requests on LAN interfaces
- Send out SLARP requests on WAN interfaces
- On built-in Fast Ethernet interfaces:
 - Attempt DHCP and RARP address acquisition 3 times
 - Assign a static IP if dynamic acquisition failed
 - Start a DHCP server if dynamic acquisition failed
 - Accommodates initial configuration using J-Web



LAN Interfaces

LAN interfaces use the Dynamic Host Configuration Protocol (DHCP) and the Reverse Address Resolution Protocol (RARP) to obtain an IP address.

WAN Interfaces

Point-to-point WAN interfaces use the Serial Line Address Resolution Protocol (SLARP) over Cisco-HDLC encapsulation for address acquisition.

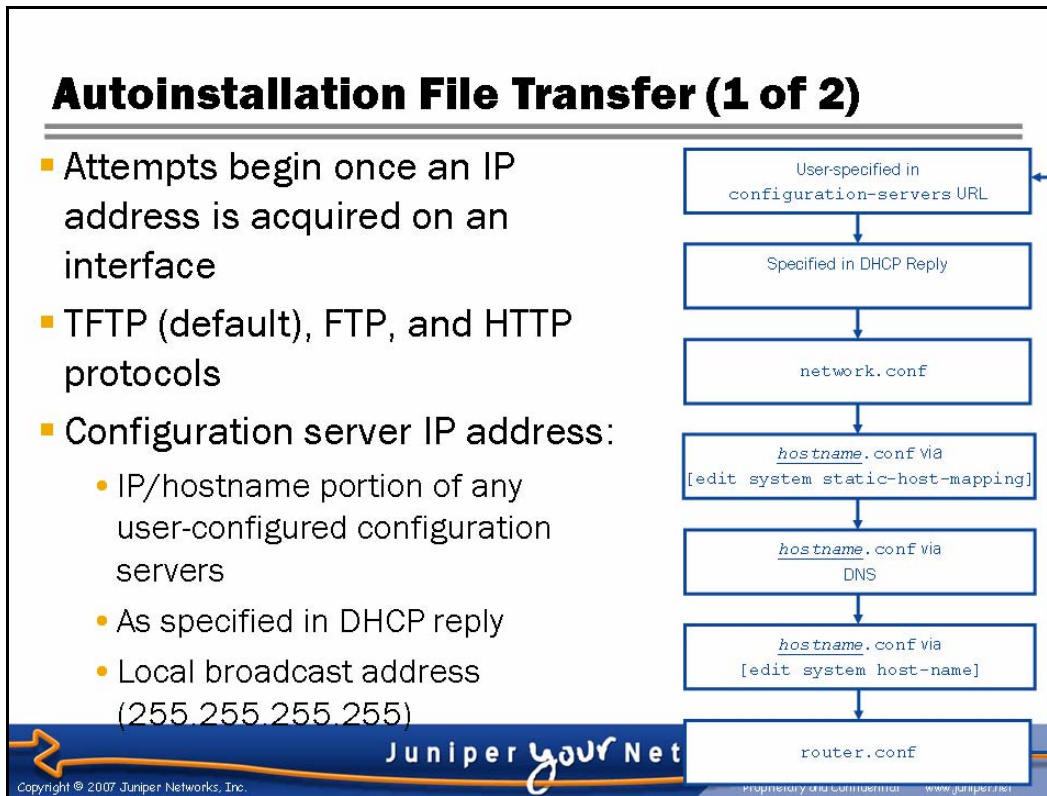
Continued on next page.

DHCP Server Mode

Each of the built-in Fast Ethernet or Gigabit Ethernet interfaces will attempt DHCP/RARP address acquisition three times. If they do not acquire an address after three attempts, they are assigned a static address and start a DHCP server process on the interface. This process accommodates quick initial configuration using the J-Web interface from a directly attached management host.

The `fe-0/0/0` or `ge-0/0/0` interface is assigned a static IP address of 192.168.1.1/24, and the corresponding DHCP server process assigns IP addresses between 192.168.1.2/24 and 192.168.1.254/24 with a 12-hour lease time. The `fe-0/0/1` or `ge-0/0/1` interface is assigned a static IP address of 192.168.2.1/24, and the corresponding DHCP server process assigns IP addresses between 192.168.2.2/24 and 192.168.2.254/24 with a 12-hour lease time.

This DHCP server process stops once autoinstallation is complete. JUNOS software Release 7.2 and later also support a configurable DHCP server process for post-installation IP address assignment to LAN clients.



Autoinstallation Sequence

Attempts to transfer a configuration file begin once an IP address is acquired on an interface. An IP address can be acquired through static configuration, an address resolution protocol, or the static addresses (192.168.1.1 and 192.168.2.1) used for DHCP server mode.

Supported File Transfer Protocols

By default, the router attempts to transfer the file using the TFTP protocol only. You can specify the URL of a configuration server at the `[edit system autoinstallation configuration-servers]` hierarchy. These URLs support TFTP, FTP, and HTTP.

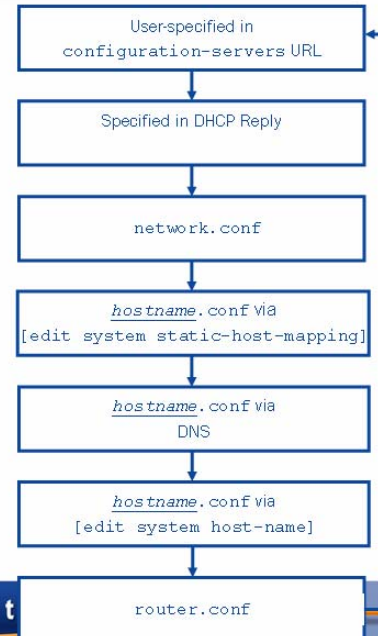
Determining the Configuration Server IP Address

You can specify the IP address or hostname of the configuration server by configuring a URL at the `[edit system autoinstallation configuration-servers]` hierarchy. If there is no user-configured URL, any configuration server IP address specified in the DHCP reply is used. If no configuration server IP address was specified in the DHCP reply, the local broadcast address of 255.255.255.255 is used.

Autoinstallation File Transfer (2 of 2)

■ Configuration file name:

- File name portion of any user-configured configuration servers
- As specified in DHCP reply
- `network.conf`
- `hostname.conf`, where `hostname` is resolved via:
 - `[edit system static-host-mapping]`
 - DNS
 - `[edit system host-name]`
- `router.conf`



Determining the Configuration File Name

The autoinstallation process continuously attempts to transfer configuration files of various names until a file is successfully transferred and committed. JUNOS software uses the following steps determine the name and order of the requested configuration files:

1. If a configuration server's URL is configured at the `[edit system autoinstallation]` hierarchy, and the URL contains a filename portion, request the specified filename.
2. If a DHCP reply was received that included a configuration filename, request that filename.
3. Request a file named `network.conf`.
4. Request a file named `hostname.conf` where `hostname` is determined by the `[edit system static-host-mapping]` hierarchy configuration.
5. Request a file named `hostname.conf` where `hostname` is resolved via DNS.
6. Request a file named `hostname.conf` where `hostname` is resolved by the `[edit system host-name]` configuration.
7. Request a file named `router.conf`.

Monitoring and Controlling Autoinstallation

- CLI commands control and monitor the autoinstallation process:
 - **show system autoinstallation status**
 - **request system autoinstallation stop** (hidden)
 - **restart autoinstallation** (hidden)

```

root> show system autoinstallation status
Autoinstallation status:
Master state: Active
Last committed file: None
Configuration server of last committed file: None
Interface:
  Name: fe-0/0/0
  State: End
  Acquired:
    Address: 10.0.1.69
    Hostname: None
    Hostname source: None
    Configuration filename: None
    Configuration filename server: 255.255.255.255
  Address acquisition:
    Protocol: DHCP Client
    Acquired address: None
    Protocol: RARP Client
    Acquired address: None
    . . .
    
```

No configuration file is committed

Static IP address assignment prevents DHCP server mode on this interface



Controlling and Monitoring Autoinstallation

A set of commands allows you to control and monitor autoinstallation. The **show system autoinstallation status** command gives an overview of the autoinstallation status and whether or not a configuration file was successfully committed. It also displays the assigned IP address, configuration server, and configuration file for each interface.

The **request system autoinstallation stop** and the **restart autoinstallation** commands provide a way of stopping and starting the autod process. To avoid having inexperienced users inadvertently disrupting the autoinstallation process, these commands are hidden.

Tracing Autoinstallation

- Tracing is similar to debug
 - Requires configuration of a (hidden) traceoptions stanza under [edit system autoinstallation] hierarchy
 - Results are written to /var/log/autod
 - Monitor with **show log autod** or **monitor start autod**

```
[edit system autoinstallation]
root# show
traceoptions {
  level verbose;
  flag {
    all;
  }
}
root> show log autod
. . .
Jul 20 13:42:05 starting interface state machine for fe-0/0/0
Jul 20 13:42:05 allocated 1372 bytes at 0x8123800
Jul 20 13:42:05 starting interface state machine for fe-0/0/1
Jul 20 13:42:05 started all interface state machines
Jul 20 13:42:05 interface fe-0/0/0 now in state Link Detect
Jul 20 13:42:05 interface fe-0/0/0 now in state IFL Bringup
Jul 20 13:42:05 interface fe-0/0/0 now in state Configuration Acquisition
Jul 20 13:42:05 autod_cs_getconfigfilename: using network.conf as candidate configuration
file name
. . .
```

The desired detail level

What events should be traced



Troubleshooting Autoinstallation

Most configuration hierarchies within the JUNOS software allow traceoptions to be configured. This feature is similar to debug functionality on other vendors' equipment. Tracing sends syslog messages to a feature-specific log file where you can use the **monitor start** command to monitor the log file in real time or the **show log** command to view the file one page at a time.

The traceoptions stanza under the [edit system autoinstallation] hierarchy is hidden. When configured, it logs messages to the /var/log/autod file.

Autoinstallation Factory Defaults

- Autoinstallation is enabled in a factory-default configuration
 - Uses the hidden command **delete-upon-commit** to deactivate autoinstallation upon a successful commit
 - Configured tracing (hidden) to assist in fault analysis

```
[edit]
user@host# load factory-default
warning: activating factory configuration

[edit]
user@host# show system autoinstallation
delete-upon-commit; ## Deletes [system autoinstallation] upon change/commit
traceoptions {
    level verbose;
    flag {
        all;
    }
}
```



Factory-Default Settings

The factory-default configuration has autoinstallation enabled and uses the hidden **delete-upon-commit** command to deactivate autoinstallation upon a successful commit. The hidden **traceoptions** command is also configured to assist in autoinstallation troubleshooting. All autoinstallation logs are sent to the `/var/log/autod` file.

Agenda: Installation and Initial Configuration

- Installation Guidelines
- Autoinstallation
- ➔ Rescue and Factory-Default Configurations
- Configuration Checklist
- Initial Configuration Using J-Web
- Initial Configuration Using the CLI
- Overview of Interface Configuration
- Configuring Interfaces Using J-Web



Rescue and Factory-Default Configurations

The slide highlights the topic we discuss next.

The Rescue Configuration

- A rescue configuration is designed to restore basic connectivity in the event of configuration problems
 - Contents are user defined
 - Include a root password!
 - By default, there is no rescue configuration
 - Can be saved using J-Web or the CLI
 - Once saved, the rescue configuration can be activated with the CLI or, on the J-series routers, a momentary push of the recessed RESET CONFIG button



RESET CONFIG button

What Is a Rescue Configuration?

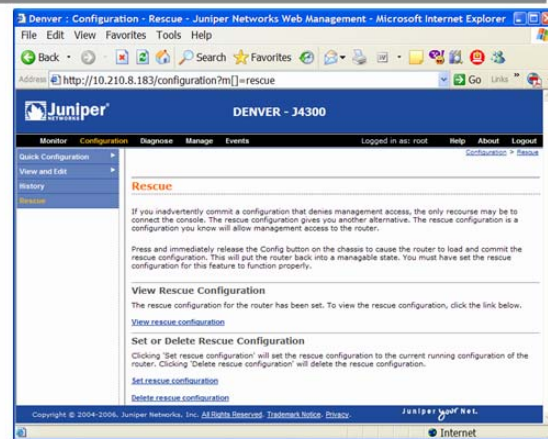
A rescue configuration is a user-defined, known-good configuration that can be quickly activated in the event that the active configuration is deleted or misconfigured in such a way that network connectivity to the router is lost. We recommend that the rescue configuration contain the minimum elements necessary to restore network connectivity to the router. For added security, the rescue configuration should include a root password.

By default, no rescue configuration is defined. You can save the current active configuration as the rescue configuration using J-Web or the CLI.

Once saved, you can activate the rescue configuration using the CLI or, on the J-series routers, by using the recessed RESET CONFIG button on the front of the chassis.

Saving a Rescue Configuration

- Use J-Web Configuration > Rescue page to view, save, or delete a rescue configuration
- Or, use request system configuration rescue [save | delete] CLI command
 - View with the show system configuration rescue CLI command



Using J-Web

The J-Web Configuration > Rescue page allows you to view, save, or delete the rescue configuration. The Set rescue configuration link sets the rescue configuration to the currently active configuration. The Delete rescue configuration link removes any rescue configuration previously set. The View rescue configuration link allows you to view the contents of the rescue configuration. It appears only if a rescue configuration is set.

Using the CLI

You can also set or delete the rescue configuration from the CLI. The **request system configuration rescue save** command sets the rescue configuration to the currently active configuration, and the **request system configuration rescue delete** command deletes any rescue configuration previously set. The **show system configuration rescue** command allows you to see the contents of the rescue configuration file, or it notifies you if no rescue configuration is set.

Loading the Rescue Configuration

- Briefly push recessed front-panel CONFIG button
 - Look for flashing green light to indicate a successful load
 - Do not hold the CONFIG button for > 15 seconds!
- Or, use the CLI's **rollback rescue** command in configuration mode
 - Remember to activate the rescue configuration with a **commit**!

```
[edit]
user@host# rollback ?
Possible completions:
  <[Enter]>      Execute this command
  0              2005-05-29 10:23:01 UTC by user via cli
  1              2005-05-29 09:58:12 UTC by user via cli
  2              2005-05-27 17:38:15 UTC by user via cli
  ...
  49            2005-04-21 04:49:29 UTC by root via autoinstall
  rescue        2005-05-29 11:43:31 UTC by user via cli
  |             Pipe through a command
[edit]
user@host# rollback rescue
load complete
```

Activates rescue configuration



The RESET CONFIG Button

You can activate the rescue configuration by *briefly* pressing the recessed RESET CONFIG button on the front of the chassis. Holding the RESET CONFIG button for longer than 15 seconds deletes the active configuration, the rescue configuration, all rollback configurations, and activates a factory-default configuration! Briefly pressing the RESET CONFIG button when no rescue configuration is set has no effect.

On the J2300 router, the configuration LED next to the CONFIG button indicates status. This LED blinks green while the rescue configuration is being loaded. It lights steadily green when the rescue configuration or factory-default configuration is loaded and committed. The configuration LED blinks red while all configurations are being deleted and the factory-default configuration is being loaded and committed. The configuration LED lights steadily red if a recovery operation fails.

Using **rollback rescue**

The configuration-mode **rollback** command also accepts a rescue argument. Using **rollback rescue** overwrites the candidate configuration with the rescue configuration. As always, you must use the **commit** command to activate the candidate configuration.

The Factory-Default Configuration

```
[edit]
user@host# show
system {
    autoinstallation {
        delete-upon-commit; ## Deletes [system autoinstallation] upon change/commit
        traceoptions {
            level verbose;
            flag {
                all;
            }
        }
    }
    services {
        web-management {
            http;
        }
    }
    syslog {
        user * {
            any emergency;
        }
        file messages {
            any any;
            authorization info;
        }
        file interactive-commands {
            interactive-commands any;
        }
    }
}
```

A factory-default configuration supports root user logins only

By default, the root user has no password and can only log in using the console or J-Web



The Factory-Default Configuration

As previously discussed, the factory-default configuration enables autoinstallation and J-Web access using HTTP. It also enables basic system logging and autoinstallation tracing for troubleshooting purposes.

No user accounts or network configuration is included. Thus, only the default root user can log in using the console or J-Web. By default, the root user has no password.

Returning to a Factory Configuration


- There might be times when you want to return to a factory configuration
 - Reactivating autoinstallation, etc.
- Use the `load factory-default` CLI configuration-mode command and set a root password

```
[edit]
lab@Denver# load factory-default
warning: activating factory configuration

[edit]
lab@Denver# set system root-authentication plain-text-password
New password:
Retype new password:

[edit]
lab@Denver# commit
commit complete
```

Activates the factory-default configuration



- Depress the front-panel CONFIG button for >15 seconds
 - This method deletes the active configuration, the rescue configuration, and all rollback configurations!
 - Unit should be secured to prevent access to the CONFIG button



Returning to a Factory-Default Configuration

Under certain conditions, you might want to return the router to its factory-default configuration. For example, you might want to reactivate autoinstallation or simply clear the configuration to prepare the router for redeployment in a new role.

Using the CLI

The CLI's configuration mode allows you to overwrite the candidate configuration with the factory-default configuration by using the `load factory-default` command. Recent JUNOS software versions do not allow you to save the configuration until you configure root authentication information. Do not forget to issue a `commit` to activate your changes.

Continued on next page.

Using the RESET CONFIG Button

You can also use the RESET CONFIG button on the front of the chassis to load a factory-default configuration. You perform this process by pressing the recessed RESET CONFIG button for more than 15 seconds. This method, however, will also delete all configuration files including the current active configuration, the rescue configuration, and all rollback configurations.

On the J2300 router, the configuration LED next to the CONFIG button indicates status. This LED lights steadily green when the rescue configuration or factory-default configuration is loaded and committed. The configuration LED blinks red while all configurations are being deleted and the factory-default configuration is being loaded and committed. The configuration LED lights steadily red if a recovery operation fails.

Physical security of the router prevents accidental or malicious access to the RESET CONFIG button. Of course, physical security is important for all sensitive equipment—not just J-series routers. Although it is not a substitute for adequate physical security, you can also disable the use of the RESET CONFIG button for loading the factory-default configuration, the rescue configuration, or both. You can disable these under the `[edit chassis config-button]` hierarchy.

Agenda: Installation and Initial Configuration

- Installation Guidelines
- Autoinstallation
- Rescue and Factory-Default Configurations
- ➔ Configuration Checklist
- Initial Configuration Using J-Web
- Initial Configuration Using the CLI
- Overview of Interface Configuration
- Configuring Interfaces Using J-Web



Configuration Checklist

The slide highlights the topic we discuss next.

Initial Configuration Checklist

- These items are normally configured (as needed) using the J-Web quick configuration setup wizard or the CLI:
 - Root password
 - Hostname
 - System time
 - Domain name and DNS server address
 - Remote access protocols
 - Management and loopback interface properties
 - A default route



Initial Configuration

When you receive a Juniper Networks J-series router, the JUNOS software is preinstalled. Once you power on the router, it is ready to attempt autoinstallation or it is ready for manual configuration. For the latter, you can configure the router from a console connected to the router's console port or using J-Web from a management host directly attached to one of the built-in Fast Ethernet or Gigabit Ethernet interfaces. We recommend you configure the following items at installation time:

- Root password (By default, the only user that can access a router is root. There is no root password specified in the initial active configuration, so we recommend setting this password immediately.);
- Hostname of the router;
- Time of day/Network Time Protocol;
- Domain name and IP address of a Domain Name System (DNS) server;
- System services for remote access (Telnet, SSH, and HTTP/HTTPS);
- Management interface IP address (While J-series units do not have a dedicated management interface, it is good practice to reserve the fe-0/0/0 or ge-0/0/0 interface for out-of-band management network usage.);
- Loopback interface, and
- IP address of a default router for the management network.

Secondary Configuration Checklist

- After initial configuration use the various quick configuration wizards to configure:
 - User accounts and permissions
 - SNMP network management
 - Interface properties
 - Routing protocols, firewall filters, NAT, etc.
- All in due time...
 - The configuration of routing protocols, firewall filters, and services are covered in subsequent modules



Secondary Configuration

After initial configuration you normally move on to secondary items that include the following:

- Local user accounts;
- SNMP network management;
- Loopback and transient interfaces; and
- Any remaining functionality needed to place the router into service, for example, routing protocols, routing policy, firewall filters, etc.

Stay Tuned

The last grouping of items are detailed in their respective section in upcoming chapters.

Agenda: Installation and Initial Configuration

- Installation Guidelines
- Autoinstallation
- Rescue and Factory-Default Configurations
- Configuration Checklist
- ➔ Initial Configuration Using J-Web
- Initial Configuration Using the CLI
- Overview of Interface Configuration
- Configuring Interfaces Using J-Web



Initial Configuration Using J-Web

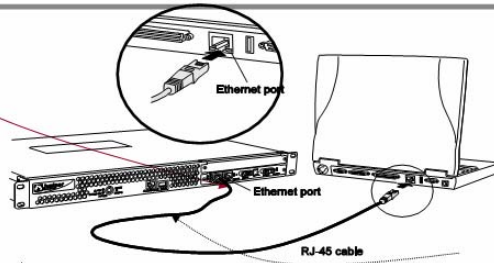
The slide highlights the topic we discuss next.

Accessing J-Web with Factory Defaults

DHCP server mode on built-in FE interfaces only

■ Preferred approach:

1. Attach a PC with 128-bit browser support directly to the `fe-0/0/0` or `ge-0/0/0` interface with an RJ-45 Ethernet cable
 - PC can be configured for DHCP or static IP address in the form of `192.168.1.x/24`, where `x` is any value from 2 to 254, inclusive
2. Wait for DHCP server mode to activate after three unsuccessful attempts to obtain a dynamic IP address
3. Point your browser to `http://192.168.1.1`
4. You are automatically logged in as root and presented with the Configuration > Quick Configuration > Setup wizard



Initial Configuration with J-Web

Thanks to the default autoinstallation behavior, using the J-Web interface for initial configuration is extremely easy. Simply use an RJ-45 cable to directly connect the DHCP-configured management host to one of the router's two built-in Fast Ethernet interfaces. The router will configure the `fe-0/0/0` or `ge-0/0/0` interface with an IP address of `192.168.1.1` and the `fe-0/0/1` or `ge-0/0/1` interface with an IP address of `192.168.2.1`. It also acts as a DHCP server on these interfaces, assigning IP addresses in the `192.168.1.0/24` and `192.168.2.0/24` networks, respectively. IP addresses assigned by the DHCP server have a 12-hour lease time.

Point the Web browser on your management host to `http://192.168.1.1` (or `http://192.168.2.1` for `fe-0/0/1` or `ge-0/0/1`.) You will be automatically directed to the J-Web Configuration > Quick Configuration > Setup wizard where you can fill in initial configuration parameters.

The J-Web Setup Wizard

- Fill in desired fields; fields flagged with * are required

The J-Web Setup Wizard

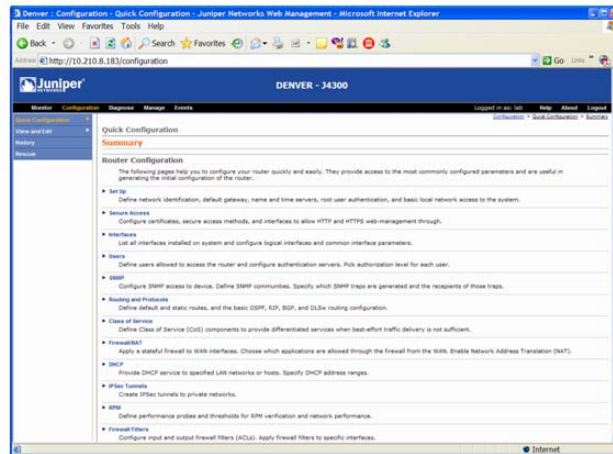
The Configuration > Quick Configuration > Setup wizard provides a single location to fill in most of the initial configuration parameters. The fields marked with a red asterisk (*) are required. Once you finish entering information, clicking OK or Apply activates your configuration. Watch for any error messages indicating the configuration was not committed.

If you are connected to J-Web through an interface and use the setup wizard to modify that interface's IP address, applying your changes will cause you to lose connectivity. You must reconnect using the new IP address.

Even if you do not change the built-in Fast Ethernet IP addresses, you should remember that committing an initial configuration halts autoinstallation's DHCP server process. If your management host is using a DHCP-assigned IP address, you will lose connectivity when your 12-hour lease expires.

The Quick Configuration Summary

- You are now presented with the Configuration > Quick Configuration > Summary page
 - Use remaining wizards to perform secondary configuration tasks



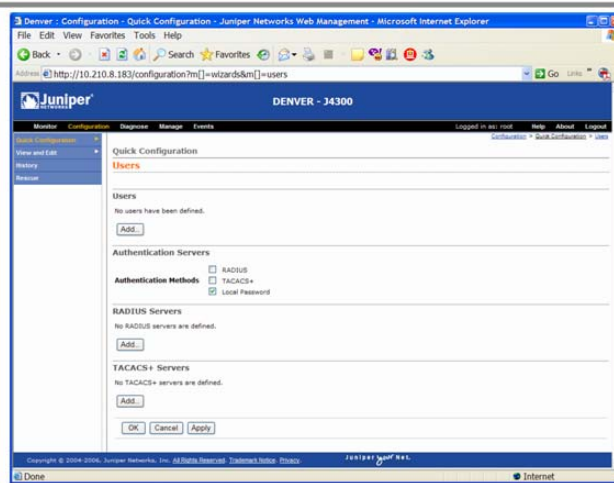
Quick Configuration Summary Page

The J-Web Configuration > Quick Configuration > Summary page provides quick access to other wizards that can speed initial configuration. These wizards include the following:

- **SSL:** Configure SSL certificates and access for J-Web over HTTPS.
- **Interfaces:** Configure transient interface physical and logical properties.
- **Users:** Define users allowed to access and associate authentication and authorization parameters.
- **SNMP:** Configure SNMP access to the router.
- **Routing:** Configure default and static routes. Perform basic configuration of dynamic routing protocols.
- **Firewall/NAT:** Configure and apply stateful firewalls and apply NAT.
- **IPsec Tunnels:** Create encrypted tunnels to form virtual private networks.
- **Real Time Performance Monitoring:** Define probes for measuring network performance.

Quick Configuration: Adding Users

- Configure user accounts and authentication settings at the Configuration > Quick Configuration > Users page

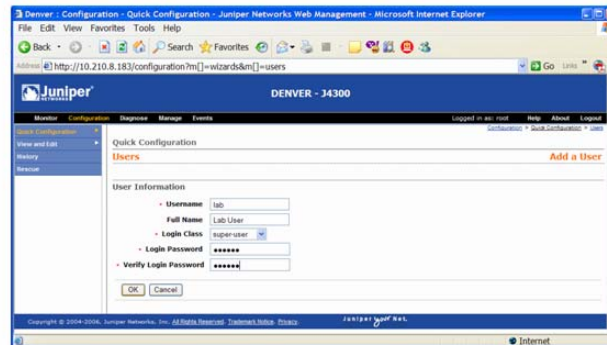


User Account and Authentication Settings

You can use the J-Web Configuration > Quick Configuration > Users wizard to create local user accounts and to define RADIUS and TACACS+ authentication servers. You can also define which authentication methods should be used. You must perform advanced configuration of authentication-order from the CLI or the J-Web Configuration > View and Edit hierarchy.

Quick Configuration: Defining Users

- Specify username, login class, and password
 - Select a predefined login class from pull-down list
 - Configure login classes at Configuration > View and Edit > Edit Configuration > System > Login, or through CLI
 - Click OK to activate



Adding Users

Add users at the Configuration > Quick Configuration > Users hierarchy by filling in the Username, Full Name, Login Class, and Login Password fields.

The Login Class box is populated with all currently defined login classes. These classes include the four predefined classes as well as any classes the user configured by using the CLI or the J-Web Configuration > View and Edit hierarchy.

Once you enter the necessary information for a new user, click OK to apply your changes and return to the Configuration > Quick Configuration > Users hierarchy.

Quick Configuration: SNMP (1 of 2)

■ Configure SNMP at the Configuration > Quick Configuration > SNMP page

Contact and description data

Click to define communities or trap groups

Health monitoring configuration

Juniper your Net

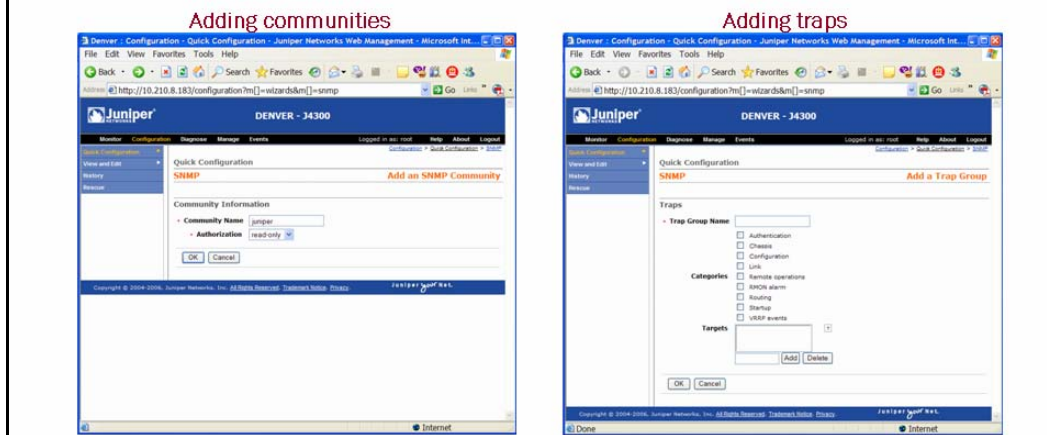
Copyright © 2007 Juniper Networks, Inc. Proprietary and Confidential www.juniper.net

SNMP Configuration Wizard

You can enter SNMP identification information at the Configuration > Quick Configuration > SNMP wizard. Click the appropriate Add... button to configure SNMP communities and SNMP traps. You can also configure the router to track certain system variables (such as CPU utilization, memory utilization, and file system utilization) and create an alarm when a threshold is crossed. Do not forget to click OK or Apply once you finish.

Quick Configuration: SNMP (2 of 2)

- Define zero or more communities and the associated authorization level
- Define zero or more trap groups
 - Include a name for the trap group, one or more categories, and one or more targets



Configuring SNMP Communities

You can enable the router to respond to SNMP requests from an SNMP manager by defining an SNMP community. A read/write community responds to SNMP get requests and SNMP set requests, while a read-only community responds only to SNMP get requests. Click OK to commit your changes.

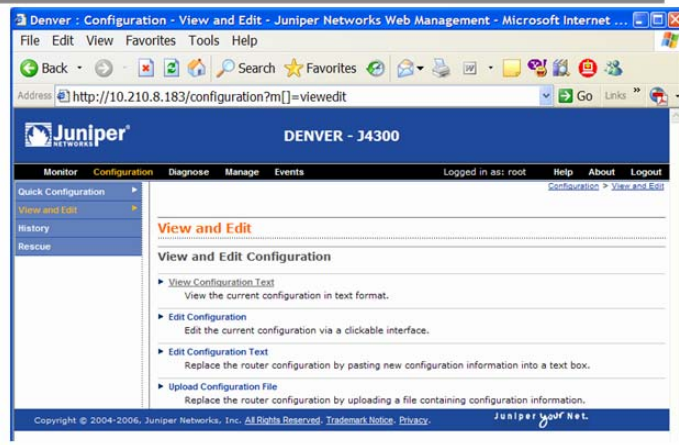
You must perform more advanced configurations, such as restricting SNMP access by IP address, with the CLI or J-Web's Configuration > View and Edit hierarchy.

Configuring SNMP Traps

The router can also send event notifications to an SNMP manager using SNMP traps. Each category defines the type of events about which you want to be notified. Targets are a list of SNMP managers to which SNMP traps will be sent. Again, click OK to commit your changes.

Displaying Initial Configuration

- Use the Configuration > View and Edit > View Configuration Text page to display your initial configuration



Reviewing Your Work

You can now use the J-Web View Configuration Text page to display the active configuration in the hierarchical syntax used by the CLI.

Agenda: Installation and Initial Configuration

- Installation Guidelines
- Autoinstallation
- Rescue and Factory-Default Configurations
- Configuration Checklist
- Initial Configuration Using J-Web
- ➔ Initial Configuration Using the CLI
- Overview of Interface Configuration
- Configuring Interfaces Using J-Web



Initial Configuration Using the CLI

The slide highlights the topic we discuss next.

Initial Configuration Using the CLI (1 of 6)

■ Log in as root

```
..  
rc.i386 configuring syscons:..  
Local package initialization:..  
starting local daemons:..  
Mon May 30 21:14:50 UTC 2005
```

Amnesiac prompt indicates a
factory-default configuration

```
Amnesiac (ttyd0)
```

```
login: root
```

```
--- JUNOS 8.1R2.4 built 2006-12-29 08:27:34 UTC
```

```
root@%
```

UNIX shell prompt

■ Start CLI

```
root@% cli  
root>
```

CLI prompt



Logging In as Root

When you receive a J-series platform from the factory, the root password is not set. To log in to the router's CLI for the first time, you must log in through the console port using the root username with no password.

The console login normally displays the router's configured hostname. When no hostname is configured, such as the case with a factory-default configuration, *Amnesiac* is displayed in place of the hostname.

Starting the CLI

When you log in as the root user, you are placed at the UNIX shell. You must start the CLI by typing the **cli** command. When you exit the CLI, you return to the UNIX shell. For security reasons, make sure you also log out of the shell using the **exit** command.

Initial Configuration Using the CLI (2 of 6)

- Enter configuration mode

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

- Issue CLI commands to configure desired functionality

- Remember to issue the **commit** command to activate your changes
- Hint: Use the CLI's | **display set** functionality to reverse-engineer a configuration into the CLI commands used to create it



Entering Configuration Mode

After starting the CLI, you enter operational mode. You can make changes to the configuration only in configuration mode. Enter configuration mode by entering the command **configure** at the operational-mode prompt, as shown on the slide.

Continued on next page.

Issuing Commands

Once in configuration mode you issue **set** commands to configure the desired functionality. Remember that your changes do not take effect until you issue a **commit** command. To help learn CLI configuration syntax, you might try displaying a configuration with the results piped to the `display set` functionality as shown:

```
[edit]
```

```
root# show interfaces lo0
```

```
unit 0 {  
    family inet {  
        address 10.0.0.1/32;  
    }  
}
```

```
[edit]
```

```
root# show interfaces lo0 | display set
```

```
set interfaces lo0 unit 0 family inet address 10.0.0.1/32
```

Initial Configuration Using the CLI (3 of 6)

- Set the identification parameters

- Hostname
- Domain name
- Root password

```
[edit]
root# edit system

[edit system]
root# set host-name host

[edit system]
root# set domain-name example.com

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

[edit system]
root#
```



Identification Parameters

This slide shows how to use the CLI to configure the same information as the identification section of the J-Web Configuration > Quick Configuration > Set Up wizard.

Initial Configuration Using the CLI (4 of 6)

- Set the time parameters

- Time zone
- NTP server
- Current time

```
[edit system]
root# set time-zone America/Los_Angeles
```

```
[edit system]
root# set ntp boot-server 10.0.3.1
```

```
[edit system]
root# set ntp server 10.0.3.1
```

```
[edit system]
root# run set date 200505280900.00
Sat May 28 09:00:00 UTC 2005
```



Identification Parameters

This slide shows how to use the CLI to configure the same information as the time section of the J-Web Configuration > Quick Configuration > Set Up wizard.

Initial Configuration Using the CLI (5 of 6)

- Set the network parameters
 - DNS name servers
 - Domain search
 - Default gateway
 - Loopback address
 - fe-0/0/0 or ge-0/0/0 address

```
[edit system]
root# set name-server 10.0.2.1

[edit system]
root# set domain-search example.com

[edit system]
root# top

[edit]
root# set routing-options static route 0.0.0.0/0 next-hop 10.0.1.254

[edit]
root# set interfaces lo0 unit 0 family inet address 10.0.0.1/32

[edit]
root# set interfaces fe-0/0/0 unit 0 family inet address 10.0.1.1/24
```



Identification Parameters

This slide shows how to use the CLI to configure the same information as the network section of the J-Web Configuration > Quick Configuration > Set Up wizard.

Initial Configuration Using the CLI (6 of 6)

- Set the management access parameters
 - Telnet
 - SSH
- Commit the changes!

```
[edit]
root# edit system

[edit system]
root# set services telnet

[edit system]
root# set services ssh

[edit system]
root# commit and-quit
May 28 02:03:18 init: autoinstallation (PID 2460) exited with status=0
Normal Exit
Commit complete
Exiting configuration mode

root@host>
```



Identification Parameters

This slide shows how to use the CLI to configure the same information as the management access section of the J-Web Configuration > Quick Configuration > Set Up wizard.

Applying Your Configuration

Now that you have completed your initial configuration, use the **commit** command to apply your changes. You can include the **and-quit** option, as shown, to return operational mode.

Note that the autoinstallation process activated by the factory-default configuration exits due to the **delete-upon-commit** statement included in the factory-default configuration.

This slide shows how to use the CLI to configure the same information as the network section of the J-Web Configuration > Quick Configuration > Set Up wizard.

Initial Configuration Results (1 of 2)

```

root@host> show configuration
version 8.1R2.4;
system {
  host-name host;
  domain-name example.com;
  domain-search example.com;
  time-zone America/Los_Angeles;
  root-authentication {
    encrypted-password "$1$VEHi2fQx$nosjW.0E9aH2mBZqFFJ7z/"; ## SECRET-DATA
  }
  name-server {
    10.0.2.1;
  }
  services {
    ssh;
    telnet;
    web-management {
      http;
    }
  }
  syslog {
    ...
  }
  ntp {
    boot-server 10.0.3.1;
    server 10.0.3.1;
  }
}

```



Displaying the Initial Configuration: Part 1

This slide uses the operational-mode **show configuration** command to display the hierarchical configuration file created by our initial configuration **set** statements. The **syslog** hierarchy that is included in the factory-default configuration is suppressed for brevity.

Initial Configuration Results (2 of 2)

```
interfaces {
  fe-0/0/0 {
    unit 0 {
      family inet {
        address 10.0.1.1/24;
      }
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 10.0.0.1/32;
      }
    }
  }
}
routing-options {
  static {
    route 0.0.0.0/0 next-hop 10.0.1.254;
  }
}
```



Displaying the Initial Configuration: Part 2

This slide displays the remaining `interfaces` and `routing-options` hierarchies created by our initial configuration `set` statements.

Secondary Configuration

- Configure local user accounts
 - [edit system login user]
- Configure SNMP communities and traps
 - [edit snmp community *community-name*]
 - [edit snmp trap-group *trap-group-name*]

```
[edit]
root@host# show snmp
community juniper {
    authorization read-only;
}
trap-group Config_and_Auth {
    categories {
        authentication;
        configuration;
    }
    targets {
        10.0.0.1;
    }
}
```



Configuring Local User Accounts

You can provision user accounts at the [edit system login user user-name] configuration hierarchy. An example is shown here:

```
[edit]
root@host# show system login
user lab {
    class superuser;
    authentication {
        encrypted-password "$1$Xp8kWSWc$Vk.7vpeXVeOTKMd ...
    }
}
```

SNMP Configuration

You configure SNMP at the [edit snmp] top-level configuration hierarchy. The slide demonstrates the configuration of an SNMP community and trap group.

Agenda: Installation and Initial Configuration

- Installation Guidelines
- Autoinstallation
- Rescue and Factory-Default Configurations
- Configuration Checklist
- Initial Configuration using J-Web
- Configuration Using the CLI
- ➔ Overview of Interface Configuration
- Configuring Interfaces Using J-Web



Overview of Interface Configuration

The slide highlights the topic we discuss next.

J-series Interface Naming Review

- J-series interfaces follow the same three-level naming convention as used in M-series and T-series platforms
 - Based on a MM-F/P/T model, where:
 - MM = The media type (e1, fe, se, t1, t3, etc.)
 - F = The PIM slot number; fixed interfaces use slot 0, expansion slots numbered from left to right, bottom to top
 - P = The virtual PIM number, currently always set to 0
 - T = The port number
 - Interface naming example: J2300 platform with 2 x FE and 2 x T1 interfaces:



J-series Interface Naming Review

This slide reiterates the J-series interface naming convention we covered in Chapter 2.

Logical Units

t1-4/0/0.43

- Logical units are like sub-interfaces in other equipment
 - In JUNOS software, a logical unit is a/ways required
 - Also used to support multipoint technologies like Frame Relay, ATM, or VLANs
- Interface unit number is separate in meaning from the actual circuit identifier; can be any arbitrary value
 - Suggested convention is to keep them the same
- PPP/HDLC encapsulations support only one logical unit
 - Must configure unit number as zero for these encapsulations
- Multiple protocol addresses are supported on a single logical unit
 - Typing in additional addresses does not override previous address
 - Watch for multiple addresses when correcting addressing mistakes!



Logical Interfaces

Each physical interface descriptor can contain one or more logical interface descriptors. These descriptors allow you to map one or more logical (sometimes called virtual) interfaces to a single physical device. Creating multiple logical interfaces is useful for ATM and Frame Relay networks, in which you can associate multiple virtual circuits or data link layer connections with a single physical interface.

Circuit Identifier Versus Unit Number

The unit number and the circuit identifier are different in meaning. The circuit identifier identifies the logical tunnel or circuit, while the unit is used to identify a logical partition of the physical interface.

Although not required, it is generally considered best practice to keep the unit number and circuit identifier the same. This practice can greatly aid in troubleshooting when you have many logical circuits.

Point-to-Point Encapsulations

PPP and Cisco HDLC encapsulations support only a single logical interface, and its logical unit number must be zero. Frame Relay and ATM encapsulations support multiple logical interfaces, so you can configure one or more logical unit numbers.

Continued on next page.

Addressing Issues

A Juniper Networks J-series platform can have more than one address on a single logical interface. Issuing a second **set** command does not overwrite the previous address but simply adds to that address. Use of the CLI's **rename** command is an excellent way to correct addressing mistakes.

Also note that JUNOS software forms IGP adjacencies over all logical interfaces when the IGP is configured on these interfaces; this behavior is worth noting because some vendors form an adjacency only over the primary address of an interface.

Interface Properties

- **Physical properties:**
 - Clocking
 - Scrambling
 - FCS
 - MTU
 - Data link layer protocol, keepalives
 - Diagnostic characteristics
 - Local, remote, and facility loopback
 - BERT
- **Logical properties:**
 - Protocol family (`inet`, `inet6`, `iso`, `mpls`)
 - Family MTU
 - Addresses (IPv4 or IPv6 address, ISO NET address)
 - Virtual circuits (VLAN tag, DLCI, VPI/VCI)
 - Etc.



Physical Properties

The following list provides details of the interface's physical properties:

- *Clocking*: Refers to the interface clock source, either internal or external.
- *Scrambling*: Refers to payload scrambling, which can be on or off.
- *Frame check sequence (FCS)*: You can modify to 32-bit mode (the default is 16-bit mode).
- *Maximum transmission unit (MTU)*: You can vary the size from 256 to 9192 bytes.
- *Data link layer protocol, keepalives*: You can change the data link layer protocol for the particular medium type (for example, PPP to Cisco HDLC), and you can turn keepalives on or off.
- *Diagnostic characteristics*: You can enable local or remote loopbacks or set up a BERT test.

Continued on next page.

Logical Properties

The following list provides details of the interface's logical properties:

- *Protocol family*: Refers to the protocol family you want to use, such as family `iso`, `inet`, or `mpls`.
- *Addresses*: Refers to the address associated with the particular family (for example, IP address using family `inet`).
- *Virtual circuits*: Refers to the virtual circuit identifier, such as a DLCI, VPI/VCI, or VLAN tag.
- *Other characteristics*: Some other configurable options include Inverse ARP, traps, and accounting profiles.

Interface Configuration

- Generic interface configuration stanza:

```

interfaces {
  interface-name {
    physical-properties;
    [...]
    unit unit-number {
      logical-properties;
      [...]
    }
  }
}

```

Physical and logical properties are configured at their respective levels

- Interface configuration example:

```

[edit]
user@host# show interfaces se-1/0/1
encapsulation cisco-hdlc;
unit 0 {
  family inet {
    address 10.222.2.2/24;
  }
}

```

Generic Interface Configuration

All interfaces have the same general configuration hierarchy organization. JUNOS software considers all properties defined directly under the interface name to be the physical properties of that interface. The unit number represents a particular logical interface or subinterface. JUNOS software considers all properties defined directly under the unit number to be the logical properties of each particular subinterface.

Agenda: Installation and Initial Configuration

- Installation Guidelines
- Autoinstallation
- Rescue and Factory-Default Configurations
- Configuration Checklist
- Initial Configuration Using J-Web
- Configuration Using the CLI
- Overview of Interface Configuration
- ➔ **Configuring Interfaces Using J-Web**

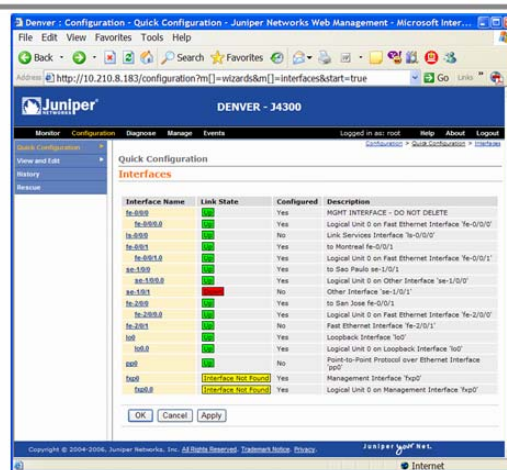


Configuring Interfaces Using J-Web

The slide highlights the topic we discuss next.

The Interface Wizard

- **Access at**
Configuration >
Quick Configuration
> Interfaces
 - The main page provides a summary of interface state and configuration status
 - Wizard currently supports IPv4 only



J-Web Interface Configuration Wizard

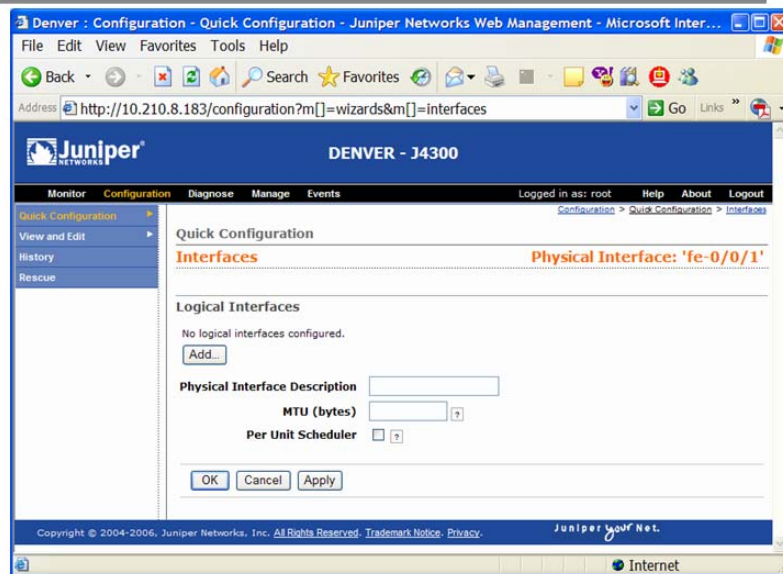
The J-Web Configuration > Quick Configuration > Interfaces wizard provides a mechanism for quickly and easily configuring basic physical and logical interface properties. Click a physical or logical interface link to proceed with configuration.

The main page of the interface wizard displays a summary of interface state and configuration status. The description field displays the configured description if configured. Otherwise, a generic description such as Fast Ethernet Interface 'fe-0/0/0' or Logical Unit 0 on Fast Ethernet Interface 'fe-0/0/0' is displayed.

The wizard currently supports IPv4 configuration only. If you must configure the `inet6`, `iso`, or `mpls` protocol families, you must use the CLI or J-Web's Configuration > View and Edit page.

Configuring Fast Ethernet (1 of 2)

- Begin by adding a logical unit to the physical device
 - Wizard currently supports Unit 0 applications only



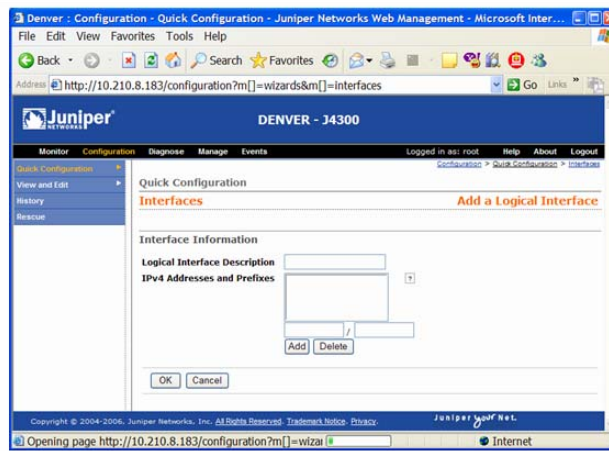
Adding a Logical Unit

You can use the J-Web interface wizard to configure a single Fast Ethernet logical interface. The J-series router supports multiple Fast Ethernet logical interfaces using 802.1q encapsulation, but this must be configured from the CLI or J-Web's Configuration > View and Edit page.

From the Configuration > Quick Configuration > Interfaces page, click the link of the Fast Ethernet you want to configure. This brings up the physical interface configuration page as shown on the slide. The only Fast Ethernet physical interface property that you can configure from the interface wizard is a description. Click the Add... button to add a logical interface and move to the logical interface configuration page.

Configuring Fast Ethernet (2 of 2)

- Specify IPv4 (family inet) addressing properties and click Add
 - You can specify additional addresses on the same unit
 - Click OK when done

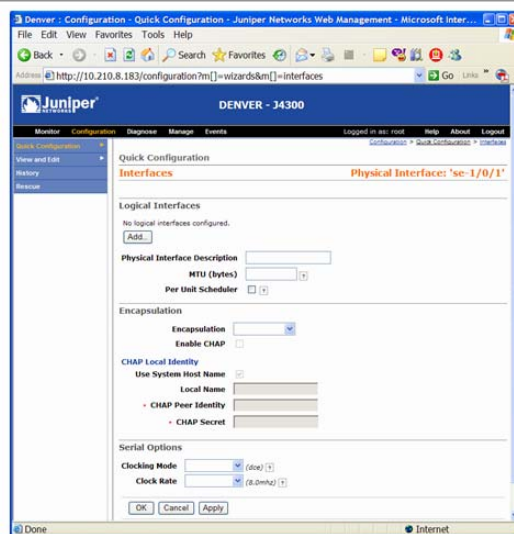


Adding an IP Address

At the logical interface page you can configure one or more IPv4 addresses. Each IP address is also configured with a subnet mask in Classless Interdomain Routing (CIDR) notation. You then click OK to apply your changes.

Configuring Serial interfaces (1 of 2)

- Configure physical properties
 - Encapsulation, MTU, clocking mode, clock rate, etc.
- Add a logical interface



Serial Interface Physical Properties

You can configure several serial interface physical properties from the interface wizard. The following list provides details of the physical properties:

- Encapsulation: Refers to data link layer protocol. All Juniper Networks point-to-point interfaces default to PPP encapsulation. Cisco HDLC and Frame Relay encapsulations are also supported.
- Enable CHAP: For PPP encapsulation only, used with the Challenge Handshake Authentication Protocol (CHAP) local identity parameters to enable CHAP authentication.
- MTU: The maximum transmission unit (MTU) specifies the maximum size, in bytes, of a packet that can be transmitted on the interface. MTU defaults to 1504 bytes on serial interfaces.
- Clocking Mode: Specifies the *source from* where the serial interface receives its clocking. The default is dce, which means the router receives its clock *from* the DCE device at the other end of the line. This setting is equivalent to line timing on other vendors' equipment.
- Clock Rate: When the router is acting as a DCE and is set to *internal* timing, the clock rate specifies the line speed in kilohertz or megahertz. Clock Rate defaults to 8.0MHz
- Per Unit Scheduler: This option is used to modify the operation of class of service (CoS) for this interface.

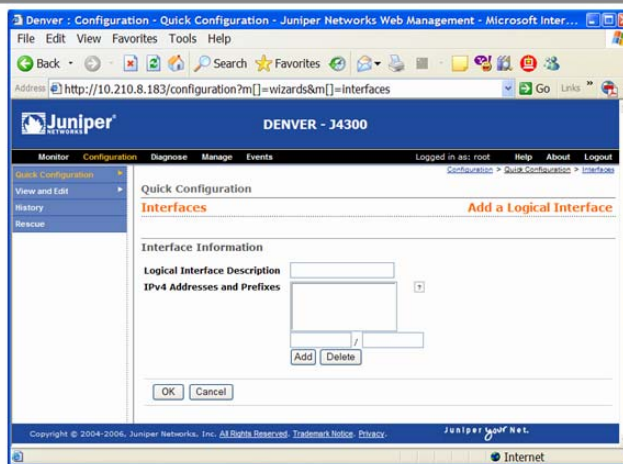
Continued on next page.

Logical Interfaces

Click the Add... button to add a logical interface. You can configure serial interfaces using Frame Relay with multiple logical interfaces from the interface wizard. PPP and Cisco-HDLC encapsulations support only a single, Unit 0, logical interface.

Configuring Serial Interfaces (2 of 2)

- Specify IPv4 (family inet) addressing properties and click Add
 - You can specify additional addresses on the same unit
 - Click OK when done

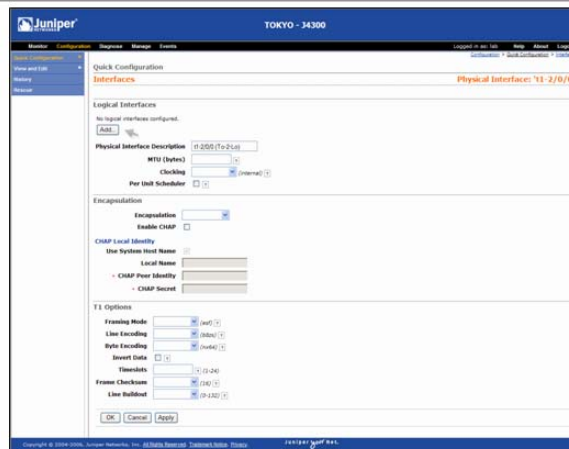


Serial Interface Logical Properties

You can configure a logical interface description and one or more IPv4 prefixes at the interface wizards logical interface page. For Frame Relay encapsulation you must also configure the DLCI corresponding to this logical interface.

Configuring T1 Interfaces (1 of 2)

- Configure physical properties
 - Encapsulation, MTU, clocking, framing, line encoding, etc.
- Add a logical interface



T1 Interface Physical Properties

You can configure several T1 physical interface properties from the interface wizard. The following list provides details of the physical properties:

- Encapsulation: Refers to data link layer protocol. All Juniper Networks point-to-point interfaces default to PPP encapsulation. Cisco-HDLC and Frame Relay encapsulations are also supported.
- Enable CHAP: For PPP encapsulation only, used with the CHAP local identity parameters to enable CHAP authentication.
- MTU: The maximum transmission unit specifies the maximum size, in bytes, of a packet that can be transmitted on the interface. MTU defaults to 1504 bytes on T1 interfaces.
- Clocking: Specifies the *source from* where the interface *receives* its clocking. The default is internal, which means the router receives clocking *from* its own system clock. Choose external to receive clocking from the T1 line.
- Framing Mode: Specifies the T1 framing mode. Extended superframe (ESF), the default, and Superframe (SF) are supported.
- Line Encoding: Specifies the line encoding method. Alternate mark inversion (AMI) and Binary 8 Zero Substitution (B8ZS) are supported. B8ZS is the default.

Continued on next page.

T1 Interface Physical Properties (contd.)

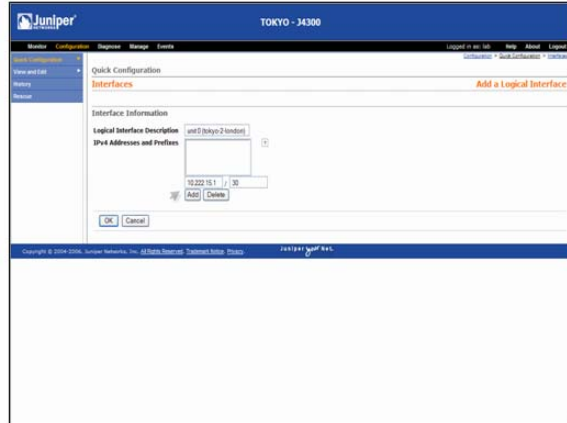
- **Byte Encoding:** Specifies the byte encoding method. nx56 uses 7 bits per byte, while nx64 is the default and uses 8 bits per byte.
- **Invert Data:** Enables or disables data inversion. Data inversion is normally used only in alternate mark inversion (AMI) mode.
- **Timeslots:** J-series T1 interfaces support fractional configurations. The timeslots parameter specifies which time slots should be allocated. By default, T1 interfaces use all time slots. You can use any combination of time slots. To configure ranges, use hyphens. To configure discontinuous slots, use commas (for example: 1-5,10,24).
- **Frame Checksum:** Specifies the number of bits in the frame checksum. A 32-bit checksum provides more reliable packet verification, but is not supported by some older equipment. The default is 16 bits.
- **Line Buildout:** Specifies the T1 cable length in feet. This parameter is used to determine how much transmit attenuation should be applied to the interface. 0–132 feet is the default.

Logical Interfaces

Click the Add... button to add a logical interface.

Configuring T1 Interfaces (2 of 2)

- Specify IPv4 (*family inet*) addressing properties and click Add
 - You can specify additional addresses on the same unit
 - Click OK when done

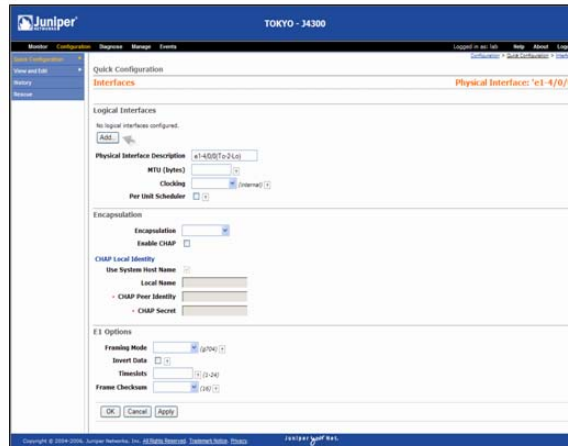


T1 Interface Logical Properties

You can configure a logical interface description and one or more IPv4 prefixes at the interface wizards logical interface page. For Frame Relay encapsulation you must also configure the DLCI corresponding to this logical interface.

Configuring E1 Interfaces (1 of 2)

- Configure physical properties
 - Encapsulation, MTU, clocking, framing, line encoding, etc.
- Add a logical interface



E1 Interface Physical Properties

You can configure several E1 physical interface properties from the interface wizard. The following list provides details of the physical properties:

- **Encapsulation:** Refers to data link layer protocol. All Juniper Networks point-to-point interfaces default to PPP encapsulation. Cisco-HDLC and Frame Relay encapsulations are also supported.
- **Enable CHAP:** For PPP encapsulation only, used with the CHAP local identity parameters to enable CHAP authentication.
- **MTU:** The maximum transmission unit specifies the maximum size, in bytes, of a packet that can be transmitted on the interface. MTU defaults to 1504 bytes on E1 interfaces.
- **Clocking:** Specifies the *source from* where the interface *receives* its clocking. The default is internal, which means the router receives clocking *from* its own system clock. Choose external to receive clocking from the E1 line.
- **Framing Mode:** Specifies the E1 framing mode. G704, with or without cyclic redundancy check 4 (CRC4), and unframed transmission formats are supported. G704 with CRC4 is the default.

Continued on next page.

E1 Interface Physical Properties (contd.)

- **Invert Data:** Enables or disables data inversion. Data inversion is normally used only in AMI mode.
- **Timeslots:** J-series E1 interfaces support fractional configurations. The timeslots parameter specifies which time slots should be allocated. By default, E1 interfaces use all time slots. You can use any combination of time slots. To configure ranges, use hyphens. To configure discontinuous slots, use commas (for example: 1-5,10,24).
- **Frame Checksum:** Specifies the number of bits in the frame checksum. A 32-bit checksum provides more reliable packet verification, but is not supported by some older equipment. The default is 16 bits.

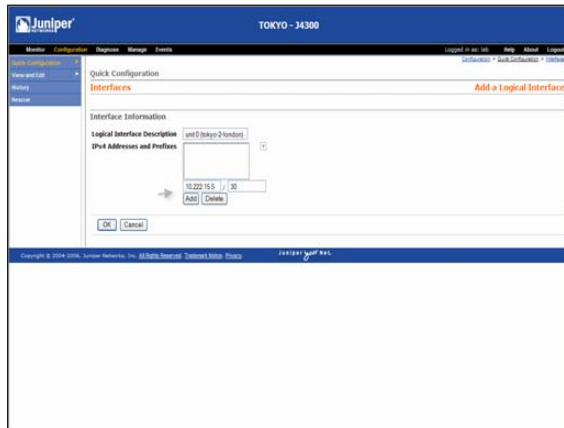
Logical Interfaces

Click the Add... button to add a logical interface.

Configuring E1 Interfaces (2 of 2)

- Specify IPv4 (*family inet*) addressing properties and click Add

- You can specify additional addresses on the same unit
- Click OK when done



E1 Interface Logical Properties

You can configure a logical interface description and one or more IPv4 prefixes at the interface wizards logical interface page. For Frame Relay encapsulation you must also configure the DLCI corresponding to this logical interface.

Configuring T3 Interfaces (1 of 2)

- Configure physical properties
 - Encapsulation, MTU, clocking, frame checksum, etc.
- Add a logical interface

T3 Interface Physical Properties

You can configure several T3 interface physical properties from the interface wizard. The following list provides details of the physical properties:

- **Encapsulation:** Refers to data link layer protocol. All Juniper Networks point-to-point interfaces default to PPP encapsulation. Cisco-HDLC and Frame Relay encapsulations are also supported.
- **Enable CHAP:** For PPP encapsulation only, used with the CHAP Local Identity parameters to enable CHAP authentication.
- **MTU:** The maximum transmission unit specifies the maximum size, in bytes, of a packet that can be transmitted on the interface. MTU defaults to 4474 bytes on T3 interfaces.
- **Clocking:** Specifies the *source from* where the T3 interface *receives* its clocking. The default is internal, which means the router receives clocking *from* its own system clock. Choose external to receive clocking from the T3 line.
- **Frame Checksum:** Specifies the number of bits in the frame checksum. A 32-bit checksum provides more reliable packet verification, but is not supported by some older equipment. The default is 16 bits.

Continued on next page.

T3 Interface Physical Properties (contd.)

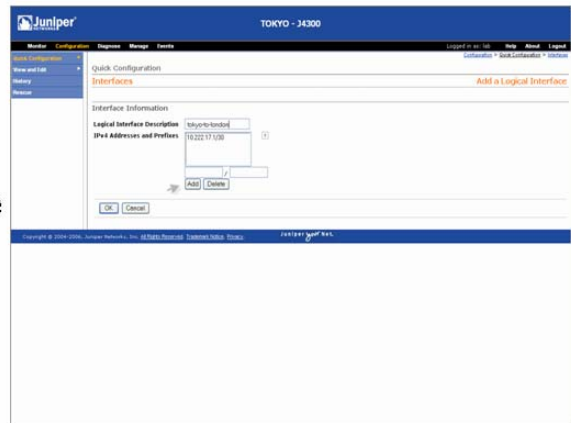
- Enable Long Buildout: This parameter is used to determine how much transmit attenuation should be applied to the interface. The box should be checked for cables longer than 225 feet.
- Disable C-bit Parity Mode: Enables or disables C-bit parity mode, which controls the type of framing that is present on the transmitted T3 signal.

Logical Interfaces

Click the Add... button to add a logical interface.

Configuring T3 Interfaces (2 of 2)

- Specify IPv4 (*family inet*) addressing properties and click Add
 - You can specify additional addresses on the same unit
 - Click OK when done



T3 Interface Logical Properties

You can configure a logical interface description and one or more IPv4 prefixes at the interface wizards logical interface page. For Frame Relay encapsulation you must also configure the DLCI corresponding to this logical interface.

Review Questions

1. Which J-series platforms can you mount on a table top?
2. What interfaces support DHCP server mode when performing autoinstallation?
3. Describe how a rescue configuration is saved and later activated.
4. How can you return a unit to a factory-default state?
5. List and describe four parameters that are normally configured during initial installation.
6. Describe parameters that are configured at the logical unit level of an interface.
7. What is the default root password?



This Chapter Discussed:

- The general process and guidelines for installing Juniper Networks routers;
- Loading a configuration file, and saving and restoring rescue configurations;
- Returning the router to its factory-default configuration; and
- Performing initial configuration on the router using both J-Web and the CLI.

Lab 2: Initial Configuration

- Perform tasks normally associated with initial installation and configuration.



Lab 2: Initial Configuration

The slide shows the objectives for this lab.



Operating Juniper Networks Routers in the Enterprise

Chapter 5: Operational Monitoring and Maintenance

Chapter Objectives

- After successfully completing this chapter, you will be able to:
 - Monitor platform and interface operation
 - Use network utilities
 - Configure system logging and parse log files for error symptoms
 - Manage licenses
 - Maintain JUNOS software
 - Perform file system maintenance and password recovery



This Chapter Discusses:

- Monitoring platform and interface operation;
- Using network utilities;
- Configuring system logging and parsing log files for error symptoms;
- Managing licenses;
- Maintaining JUNOS software; and
- Performing file system maintenance and password recovery.

Agenda: Operational Monitoring and Maintenance


- Monitoring Platform Operation
- Monitoring Interface Operation
- Network Utilities
- System Logging and Protocol Tracing
- License Management
- Maintaining JUNOS Software
- File System Maintenance and Password Recovery



Monitoring Platform Operation

The slide lists the topics we cover in this chapter. We discuss the highlighted topic first.

J-series Front Panel Indicators



- **Front panel indicators summarize platform status**
 - **Status:** Blinks green during kernel boot, steady green after boot, and blinks red on error
 - **Alarm:** On steadily red when a major alarm condition is present, on steadily yellow when a minor alarm condition is present
 - **Power:** On steadily green when powered on, blinks green when powering off

- HA: Unused
- PIM Status: PIM status LEDs vary by interface type

5-4

Copyright © 2007 Juniper Networks, Inc. Confidential www.juniper.net

J-series Status Summary

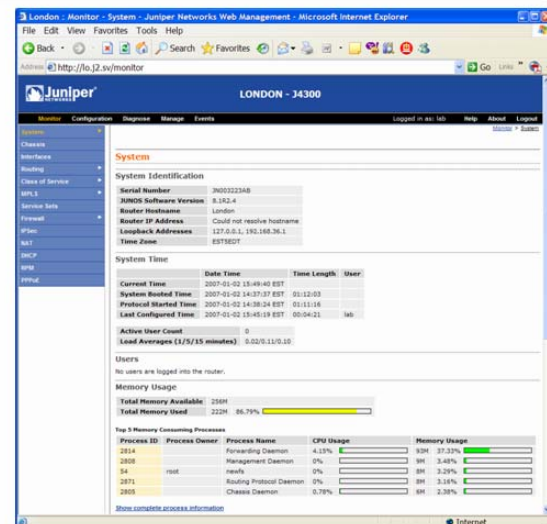
The front panel indicators on J-series platforms provide a summary of the router's status. These indicators include the following:

- **STATUS LED:** This LED blinks green while the JUNOS kernel is booting and lights steadily green after the boot up process is complete. It blinks red when an error is detected.
- **ALARM LED:** On the J4350/J6350 platforms, this LED lights steadily red to indicate a critical condition that can result in a system shutdown and lights steadily yellow to indicate a less severe condition that requires monitoring or maintenance.
- **POWER LED:** This green LED is off when the router is unplugged or is powered off and in standby mode. It lights steadily when the router is powered on and is either booting or functioning normally. The **POWER ON LED** blinks when the router is gracefully shutting down.
- **HA LED:** This LED is unused in the current JUNOS software release.
- **PIM Status LEDs:** The PIM status LEDs vary by interface type, but they usually describe the link status of the interface.

The M-series routers have analogous front-panel indicators, including indicators about which Routing Engine (RE) is active in redundant configurations. Consult the official documentation for full information about the front-panel indications for each model.

Monitoring System-Level Operation (1 of 2)

- Monitor overall system operation at the Monitor > System page
 - Or use CLI **show system** commands

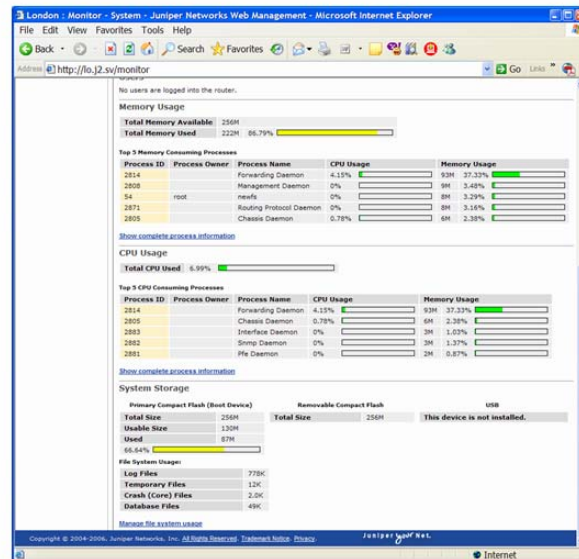


Monitoring Overall System Operation: Part 1

The J-Web Monitor > System page provides an overview of the router's operational state. You can also use various **show system** commands to retrieve equivalent information. The Monitor > System page includes the following areas:

- **System Identification:** This section shows the router's serial number, software version, hostname, and loopback IP addresses.
- **System Time:** This section displays the current time, the last time the system booted, the last time routing protocols started, and the last time the system was configured. It also includes the users currently logged in to the CLI and the load average, which is a measurement of JUNOS software utilization.
- **Users:** This section gives detailed information on users logged in to the CLI.
- **Memory Usage:** This section shows overall memory usage as well as the memory usage of important processes. The graphical displays show a green, yellow, or red bar graph that allow you to quickly identify problem areas.

Monitoring System-Level Operation (2 of 2)



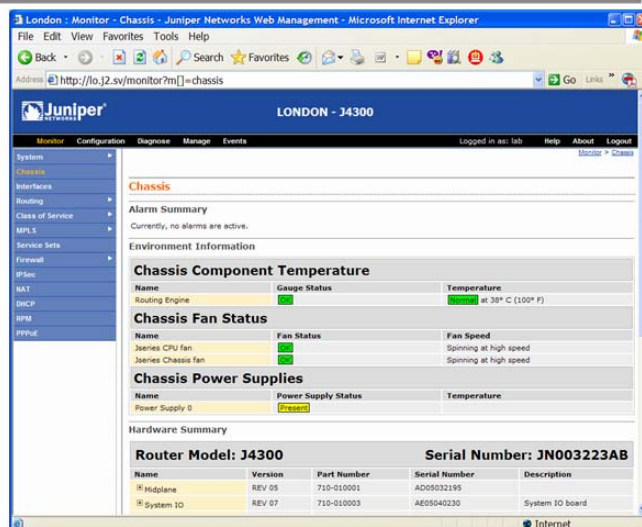
Monitoring Overall System Operation: Part 2

Additional information on the Monitor > System page includes the following areas:

- **CPU Usage:** Shows the percentage of overall processor utilization as well as the processor utilization of important processes. The graphical displays show a green, yellow, or red bar graph that allow you to quickly identify problem areas.
- **System Storage:** Shows the utilization of the primary compact flash device, removable compact flash device, and USB storage device. Again, the multicolor bar graph helps identify problem areas.

Monitoring the Chassis (1 of 2)

- Monitor chassis status at the Monitor > Chassis page
 - Or use CLI **show chassis** commands



Monitoring the Chassis: Part 1

The J-Web Monitor > Chassis page provides a convenient summary of the chassis environment. A yellow alarm is asserted when the RE temperature reaches 90 degrees Centigrade (194 degrees Fahrenheit), and a red alarm is declared when the RE temperature reaches 100 degrees Centigrade (212 degrees Fahrenheit).

Note that the chassis temperature reading as displayed with a **show chassis environment** command is actually the temperature of the air surrounding the components of the RE. To display the temperature of the CPU chip, issue a **show chassis routing-engine** command. You should expect to see that the CPU runs a bit hotter than the chassis as a whole, as shown here:

```
user@host> show chassis environment
Class Item                               Status Measurement
Temp Routing Engine                     OK           32 degrees C / 89 degrees F
Fan Fan 0                               OK
Fan Fan 1                               OK

user@host> show chassis routing-engine
Routing Engine status:
  Temperature                           32 degrees C / 89 degrees F
  CPU temperature                        37 degrees C / 98 degrees F
  DRAM                                  256 MB
  Memory utilization                     62 percent
. . .
```

Monitoring the Chassis (2 of 2)

London - Monitor - Chassis - Juniper Networks Web Management - Microsoft Internet Explorer

Address: [http://lo.j2.sv/monitor?m\[\]=chassis](http://lo.j2.sv/monitor?m[]=chassis)

Name	Power Supply Status	Temperature
Power Supply 0	Present	

Hardware Summary

Router Model: J4300 **Serial Number: JN003223AB**

Name	Version	Part Number	Serial Number	Description
Midplane	REV 05	710-010001	AD05032195	
System IO	REV 07	710-010003	AE05040230	System IO board
Routing Engine	REV 08	750-010005	BTRD44600083	RE-1.2
FPC 0				FPC
PIC 0				2x FE
FPC 1	REV 05	750-010356	AG05439232	FPC
PIC 0				2x Serial
FPC 2	REV 04	750-010353	AF06016487	FPC
PIC 0				2x FE
Power Supply 0				

FPC Summary

FPC Status

Slot	State	Temp (C)	CPU Utilization (%)		Memory DRAIN (MB)	Utilization (%)	
			Total	Interrupt		Heap	Buffer
0	Online		3	1	32	49	40
1	Online		1	1	32	26	40
2	Online		3	1	32	48	40
3	Empty						
4	Empty						
5	Empty						
6	Empty						

Copyright © 2004-2006, Juniper Networks, Inc. All Rights Reserved. Trademark Notice: Privacy

Juniper Web Net

Internet

Monitoring the Chassis: Part 2

The J-Web Monitor > Chassis page also provides a convenient summary of the hardware components installed in the router, as well as FPC status. You can access detailed information about any part in the Hardware Summary section by clicking the plus sign (+) to the left of the item.

Agenda: Operational Monitoring and Maintenance

- Monitoring Platform Operation
- ➔ Monitoring Interface Operation
- Network Utilities
- System Logging and Protocol Tracing
- License Management
- Maintaining JUNOS Software
- File System Maintenance and Password Recovery



Monitoring Interface Operation

The slide highlights the topic we discuss next.

Monitoring Interfaces

- Display summary interface status at the Monitor > Interfaces page

- Oper State and Admin State
- Physical and logical state
- Administratively disabled interface can still establish physical link

fe-0/0/0 functioning correctly

fe-0/0/1 physical interface administratively disabled

se-1/0/0 logical interface administratively disabled

se-1/0/1 data link failure

fe-2/0/0 physical link failure

fe-2/0/1 not configured

Interface Name	Oper State	Admin State	Description
fe-0/0/0	Up	Up	RIGHT INTERFACE - DO NOT DELETE
fe-0/0/1	Down	Down	
se-1/0/0	Down	Down	
se-1/0/1	Down	Down	
fe-2/0/0	Down	Down	
fe-2/0/1	Down	Down	

Summary Interface Status

The J-Web Monitor > Interfaces page provides a summary of interface status. The screen displays the operational (Oper) and administrative (Admin) state of each physical and logical interface. Unlike some vendors' equipment, administratively disabling a physical interface does not prevent the interface from establishing physical link.

On the slide interface fe-0/0/0 is up and functioning correctly. The Oper and Admin states for both the physical and logical interfaces are Up. Unless all four indicators are Up, the interface does not pass traffic.

The fe-0/0/1 physical interface is disabled. Notice that the physical interface has still established a link. The corresponding logical interface is administratively Up, but the data link layer is Down because the physical interface is disabled.

Interface se-1/0/0 demonstrates the status indicators when the logical interface is administratively disabled. Notice that the physical interface shows an operational status of Up.

A physical link failure caused interface fe-2/0/0 to display an operational state of Down for both the physical and logical interfaces. An encapsulation mis-match caused the se-1/0/1 interface to display an operational state of Down for the logical interface, while the physical interface still shows an operational status of Up.

Finally, interface fe-2/0/1 demonstrates a router interface that is simply not configured. It shows an operational state of Down because no cable is connected. Notice that it does not have a corresponding logical interface.

Displaying Terse Interface Status

- Display a summary of interface status from the CLI using the `show interfaces terse` command

```

user@host> show interfaces terse
Interface      Admin Link Proto Local                               Remote
fe-0/0/0       up    up    inet  10.251.254.3/26
...
fe-0/0/1       down  up
fe-0/0/1.0     up    down inet  10.251.254.141/31
fe-1/0/0       up    up
fe-1/0/0.0     down  up    inet  10.251.254.145/30
fe-1/0/1       up    down
fe-1/0/1.0     up    down inet  10.251.254.138/30
...
tl-4/0/0       up    down
tl-4/0/0.0     up    down inet  10.251.254.130/31
tl-4/0/1       up    down
...
lo0            up    up
lo0.0          up    up    inet  10.251.254.254    --> 0/0
                                   127.0.0.1        --> 0/0
lo0.16385      up    up    inet  10.0.0.1         --> 0/0
                                   10.0.0.16        --> 0/0
...

```



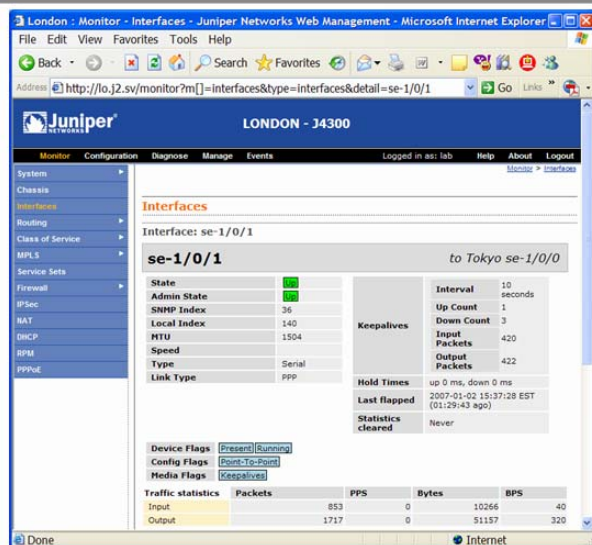
Displaying Interface Status at a Glance

Use the `show interfaces terse` command to display a terse listing of all interfaces installed in the router along with their administrative and link layer status. This output provides similar information to the J-Web Monitor > Interfaces page. The J-Web Oper status and CLI Link status are equivalent.

Note that the `lo0` interface has a logical unit of `16385` with an IP addresses of `10.0.0.1` and `10.0.0.16`, which exist in a separate routing instance named `__juniper_private1__inet.0`. The RE communicates with the virtual AS PIC through this routing instance.

J-Web Interface Details (1 of 2)

- Select an interface at the Monitor > Interfaces page to display details for that interface



J-Web Interface Details: Part 1

Clicking a physical interface at the J-Web Monitor > Interfaces page displays information about the named interface. This slide provides sample output for a J-series serial interface.

Each physical and logical interface is referenced by two index numbers within JUNOS software. A local interface index is assigned to each interface at boot time depending upon the order in which that interface is activated. The SNMP `ifIndex` is used to identify and reference that interface when performing SNMP MIB walks. Note that the indexes assigned to the physical interface device (`ifd`) differ from the index used to identify the logical device (`ifl`). Wherever possible, the SNMP `ifIndex` values are persistent across reboots or in the event of hardware additions and deletions that result from PIC or Flexible PIC Concentrator (FPC) insertion and removal. This persistence is the default behavior and is achieved by storing SNMP indexes in the `/var/db23`

`dcd.snmp_ix` file.

The top section of this J-Web page also includes the same operational and administrative states displayed on the overview page, maximum transmission unit (MTU), and media-specific configuration information such as encapsulation, loopback status, framing, and keepalives.

The next section of the Web page displays device, configuration, and media flags as well as alarm and defect information. We will now look at the possible values of each of the various flags.

Continued on next page.

J-Web Interface Details (contd.)

The possible device flags include the following:

- Down: Device is administratively disabled.
- Hear-Own-Xmit: Device will hear its own transmissions.
- Link-Layer-Down: The link layer protocol failed to successfully connect with the remote endpoint.
- Loopback: Device is in physical loopback.
- Loop-Detected: The link layer received frames that it sent and suspects a physical loopback.
- No-Carrier: Where the medium supports carrier recognition, this flag indicates that no carrier is currently seen.
- No-Multicast: Device does not support multicast traffic.
- Present: Device is physically present and recognized.
- Promiscuous: Device is in promiscuous mode and sees frames addressed to all physical addresses on the medium.
- Quench: Device is quenched because it overran its output buffer.
- Recv-All-Multicasts: No multicast filtering (multicast promiscuous).
- Running: Device is active and enabled.

The possible configuration flags include the following:

- Admin-Test: Interface is in test mode, which means that some sanity checking, such as loop detection, is disabled.
- Disabled: Interface is administratively disabled.
- Hardware-Down: Interface is nonfunctional or incorrectly connected.
- Link-Layer-Down: Interface keepalives have indicated that the link is incomplete.
- No-Multicast: Interface does not support multicast traffic.
- Point-To-Point: Interface is point to point.
- Promiscuous: Interface is in promiscuous mode and will see frames addressed to all physical addresses.
- Recv-All-Multicasts: No multicast filtering (multicast promiscuous).
- SNMP-Traps: SNMP traps are enabled.
- Up: Interface is enabled and operational.

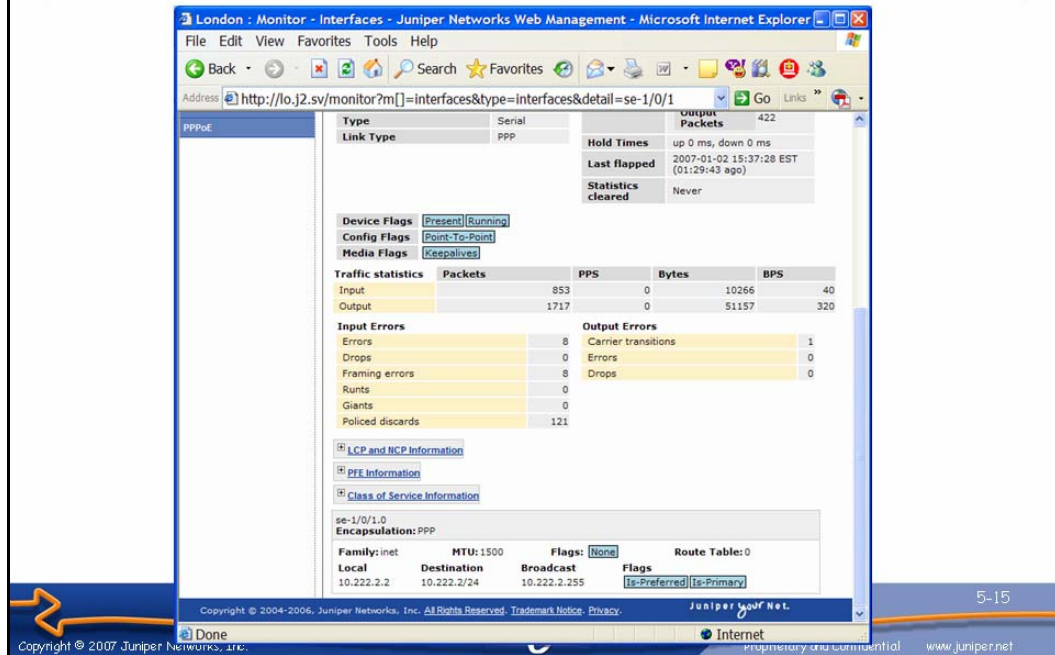
Continued on next page.

J-Web Interface Details (contd.)

The possible media flags include the following:

- Give-Up: Link protocol will not retry to connect after repeated failures.
- Keepalives: Link protocol keepalives are enabled.
- Loose-LCP: PPP will not use the Link Control Protocol (LCP) to indicate whether the link protocol is up.
- Loose-LMI: Frame Relay will not use the Local Management Interface (LMI) to indicate whether the link protocol is up.
- Loose-NCP: PPP will not use the Network Control Protocol (NCP) to indicate whether the device is up.
- No-Keepalives: Link protocol keepalives are disabled.

J-Web Interface Details (2 of 2)



J-Web Interface Details: Part 2

This slide displays the remaining sections of the J-Web interface details page. The traffic statistics sections detail the number of packets and bytes that were received and transmitted. It also includes the current packet and bit rates for both input and output.

The input and output errors sections provide counters for physical and link level errors. The following list explains the nonobvious error counters:

- **Errors:** This counter displays the sum of the incoming frame aborts and frame check sequence (FCS) errors.
- **Policed discards:** This counter displays the frames that the incoming packet match code discarded because they were not recognized or of interest. Usually, this field reports protocols that JUNOS software does not handle, such as Cisco Discovery Protocol (CDP)/Spanning Tree Protocol (STP), or any protocol type JUNOS software does not understand. (On an Ethernet network, numerous possibilities exist.)
- **L3 incompletes:** This counter is incremented when the incoming packet fails Layer 3 (usually IPv4) checks of the header. For example, a frame with less than 20 bytes of available IP header would be discarded, and this counter would be incremented.

Continued on next page.

J-Web Interface Details: Part 2 (contd.)

- L2 channel errors: This counter increments when the software cannot find a valid logical interface (such as e3-1/2/3.0) for an incoming frame.
- L2 mismatch timeouts: This counter displays the count of malformed or short packets that cause the incoming packet handler to discard the frame as unreadable.
- SRAM errors: This counter increments when a hardware error occurs in Physical Interface Module (PIM) memory. The value in this field should always be 0. If it increments, the PIM is malfunctioning.

Standard CLI Interface Display (1 of 2)

```

user@host> show interfaces t1-4/0/0
Physical interface: t1-4/0/0, Enabled, Physical link is Up
  Interface index: 142, SNMP ifIndex: 38
    Link-level type: PPP, MTU: 1504, Clocking: Internal, Speed: T1,
    Loopback: None, FCS: 16, Framing: ESF
    Device flags      : Present Running
    Interface flags: Point-To-Point SNMP-Traps 16384
    Link flags       : Keepalives
    Keepalive settings: Interval 10 seconds, Up-count 1, Down-count 3
    Keepalive: Input: 13 (00:00:05 ago), Output: 13 (00:00:01 ago)
    LCP state: Opened
    NCP state: inet: Opened, inet6: Not-configured, iso: Not-configured, mpls:
    Not-configured
    CHAP state: Not-configured
    CoS queues      : 8 supported
    Last flapped    : 2005-05-30 05:27:00 UTC (19:24:47 ago)
    Input rate      : 40 bps (0 pps)
    Output rate     : 48 bps (0 pps)
    DS1 alarms     : None
    DS1 defects    : None
    ...
  
```

Device/SNMP indexes

Device
configuration and
operational flags

Traffic loads

Standard Interface Status: Part 1

Use the **show interfaces** command without the **terse** or **detailed** switches to display standard information about the named interface (or all interfaces when a specific interface is not identified). This slide provides sample output for an J-series T1 interface. The callouts on the slide help illustrate how interfaces are partitioned into physical devices and logical units in JUNOS software.

Standard CLI Interface Display (2 of 2)

Logical interface t1-4/0/0.0 (Index 72) (SNMP ifIndex 46)

Logical device indexes

Flags: Point-To-Point SNMP-Traps Encapsulation: PPP
Protocol inet, MTU: 1500

Flags: None

Logical device settings

Addresses, Flags: Is-Preferred Is-Primary

Destination: 10.251.254.130/31, Local: 10.251.254.130

Standard Interface Status: Part 2

This slide continues the sample **show interfaces** CLI output for a J-series T1 interface.

Extensive Interface Displays (1 of 3)

- Add the **extensive** switch to display media, traffic, and error statistics
 - Use **clear interfaces** statistics to reset counters

```

user@host> show interfaces t1-4/0/0 extensive
Physical interface: t1-4/0/0, Enabled, Physical link is Up
  Interface index: 142, SNMP ifIndex: 38, Generation: 23
  Link-level type: PPP, MTU: 1504, Clocking: Internal, Speed: T1,
  Loopback: None, FCS: 16, Framing: ESF
  Device flags   : Present Running
  Interface flags: Point-To-Point SNMP-Traps 16384
  Link flags     : Keepalives
  Hold-times    : Up 0 ms, Down 0 ms
  Keepalive settings: Interval 10 seconds, Up-count 1, Down-count 3
  Keepalive statistics:
    Input : 14 (last seen 00:00:06 ago)
    Output: 14 (last sent 00:00:04 ago)
  LCP state: Opened
  NCP state: inet: Opened, inet6: Not-configured, iso: Not-configured, mpls:
    Not-configured
  CHAP state: Not-configured
  CoS queues   : 8 supported
  Last flapped : 2005-05-30 05:27:00 UTC (19:24:58 ago)
  Statistics last cleared: Never
  Traffic statistics:
    Input bytes   : 660992      0 bps
    Output bytes  : 728028      0 bps
    Input packets : 19963       0 pps
    Output packets: 20700       0 pps

```

When counters were last cleared

Traffic counters

5-19

Copyright © 2007 Juniper Networks, Inc. Proprietary and Confidential www.juniper.net

Displaying Extensive Information: Part 1

Use the **show interfaces extensive** CLI command to display detailed statistics, including traffic counters, on the interface. Use the **clear interfaces statistics interface-name** command to reset the counters for the specified interface; use the keyword **all** instead of the interface name to clear all interface statistics.

Extensive Interface Displays (2 of 3)

```

Input errors:
Errors: 0, Drops: 0, Framing errors: 0, Policed discards: 3611,
L3 incompletes: 0, L2 channel errors: 0, L2 mismatch timeouts: 0,
HS link CRC errors: 0, SRAM errors: 0, Resource errors: 0

Output errors:
Carrier transitions: 3, Errors: 0, Drops: 0, Aged packets: 0, MTU errors: 0,
Resource errors: 0

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      20693           20693           0

DS1  alarms   : None
DS1  defects  : None
Tl media:      Seconds      Count  State
SEP           0            0  OK
BEE           0            0  OK
AIS           0            0  OK
LOF          15            2  OK
LOS          15            1  OK
YELLOW        2            2  OK
BPV          17           17
EXZ           2            2
LCV           2          106
PCV           0            0
CS            0            0
LES           2            0
ES            0
SES           0
SEFS          0
BES           0
UAS           23
        
```

Input errors

Output errors

CoS Counters

Media errors

Copyright © 2007 Juniper Networks, Inc.
Proprietary and Confidential
www.juniper.net

Displaying Extensive Information: Part 2

The **show interface extensive** command also displays input and output errors, media errors, and class-of-service (CoS) counters on the interface. For a detailed description of input and output error counters, see the previous slide on J-Web interface details.

Extensive Interface Displays (3 of 3)

```

HDLC configuration:
  Policing bucket: Disabled
  Shaping bucket : Disabled
  Giant threshold: 1506, Runt threshold: 0
  Timeslots      : All active
  Line encoding: B8ZS, Byte encoding: Nx64K
  Buildout       : 0 to 132 feet
  Data inversion: Disabled, Idle cycle flag: flags, Start end flag: shared
DSL BERT configuration:
  BERT time period: 10 seconds, Elapsed: 0 seconds
  Induced Error rate: 10e-0, Algorithm: 2^15 - 1, 0.151, Pseudorandom (9)
Packet Forwarding Engine configuration:
  Destination slot: 4, PLP byte: 1 (0x00)
  CoS transmit queue

```

	%	Bandwidth bps	%	Buffer bytes	Priority	Limit
0 best-effort	95	1459200	95	0	low	none
3 network-control	5	76800	5	0	low	none

```

Logical interface tl-4/0/0.0 (Index 72) (SNMP ifIndex 46) (Generation 11)
Flags: Point-To-Point SNMP-Traps Encapsulation: PPP
Protocol inet, MTU: 1500, Generation: 16, Route table: 0
Flags: None
Addresses, Flags: Is-Preferred Is-Primary
  Destination: 10.251.254.130/31, Local: 10.251.254.130,
  Broadcast: Unspecified, Generation: 21

```

Media-specific configuration

PFE/CoS configuration

Logical interface configuration

Displaying Extensive Information: Part 3

Finally, the **show interface extensive** command provides information about media-specific configuration, Packet Forwarding Engine (PFE) and CoS configuration, and logical interface configuration.

Monitoring an Interface

- Use the **monitor interface** command for real-time statistics and error reports

```

host - SecureCRT
host                               Seconds: 132                      Time: 20:10:24
                                   Delay: 0/0/109
Interface: fe-0/0/0, Enabled, Link is Up
Encapsulation: Ethernet, Speed: 100Mbps
Traffic statistics:
  Input bytes:      973422 (73832 bps)      Current delta [520308]
  Output bytes:    2054656 (88392 bps)      [647950]
  Input packets:   11905 (110 pps)         [6329]
  Output packets:  8164 (111 pps)         [6485]
Error statistics:
  Input errors:    1095                    [0]
  Input drops:     0                      [0]
  Input framing errors: 0                  [0]
  Policed discards: 811                   [1]
  L3 incompletes:  0                      [0]
  L2 channel errors: 0                    [0]
  L2 mismatch timeouts: 0 Carrier transit [0]

Next='n', Quit='q' or ESC, Freeze='f', Thaw='t', Clear='c', Interface='i'
Ready  ssh2: AES-12  1, 78  24 Rows, 80 Cols  VT100
  
```

Monitoring an Interface

The slide depicts typical output from the **monitor interface** command. Your terminal session must support VT100 emulation for the screen to display correctly. This command provides real-time packet and byte counters as well as displaying error and alarm conditions.

Disable, Deactivate, and Bounce (1 of 2)

- Configuration-mode deactivate and disable
 - **deactivate** causes the statement or hierarchy to be ignored
 - Comments out that portion of the configuration
 - **disable** administratively disables an interface or logical unit while retaining configured properties



Configuration Mode: Deactivating and Disabling

In a configuration you can deactivate statements and identifiers so that they do not take effect when you issue the **commit** command. Any deactivated statements and identifiers are marked with the `inactive:` tag. They remain in the configuration but are not activated when you issue a **commit** command.

To deactivate a statement or identifier, use the **deactivate** configuration-mode command: **deactivate** (statement | identifier). To reactivate a statement or identifier, use the **activate** configuration-mode command: **activate** (statement | identifier). In both commands, the *statement* or *identifier* you specify must be at the current hierarchy level. While you can use the **deactivate** command on any portion of the configuration, it is especially handy to temporarily remove an interface from the configuration.

In some portions of the configuration hierarchy, you can include a **disable** statement to disable functionality. One example is disabling an interface by including the **disable** statement at the `[edit interface interface-name]` hierarchy level. When you deactivate a statement, JUNOS software completely ignores that specific object or property and does not apply it at all when you issue a **commit** command. When you disable a functionality, it is activated when you issue a **commit** command but is treated as being down or administratively disabled.

Returning to a Factory Configuration


- There might be times when you want to return to a factory configuration
 - Reactivating autoinstallation, etc.
- Use the `load factory-default` CLI configuration-mode command and set a root password

```
[edit]
lab@Denver# load factory-default
warning: activating factory configuration

[edit]
lab@Denver# set system root-authentication plain-text-password
New password:
Retype new password:

[edit]
lab@Denver# commit
commit complete
```

Activates the factory-default configuration



- Depress the front-panel CONFIG button for >15 seconds
 - This method deletes the active configuration, the rescue configuration, and all rollback configurations!
 - Unit should be secured to prevent access to the CONFIG button



Soft Booting PIMs

In some cases a soft boot or reactivating of interface hardware can recover from some failure scenarios. Rather than reboot the entire chassis, use the **request chassis fpc restart slot slot-number** command to restart a PIM. Note that this command is slightly confusing because it uses the FPC name from M-series and T-series hardware. In this case, the J-series PIM is equivalent to the M-series or T-series FPC.

Agenda: Operational Monitoring and Maintenance

- Monitoring Platform Operation
- Monitoring Interface Operation
- **Network Utilities**
- System Logging and Protocol Tracing
- License Management
- Maintaining JUNOS Software
- File System Maintenance and Password Recovery

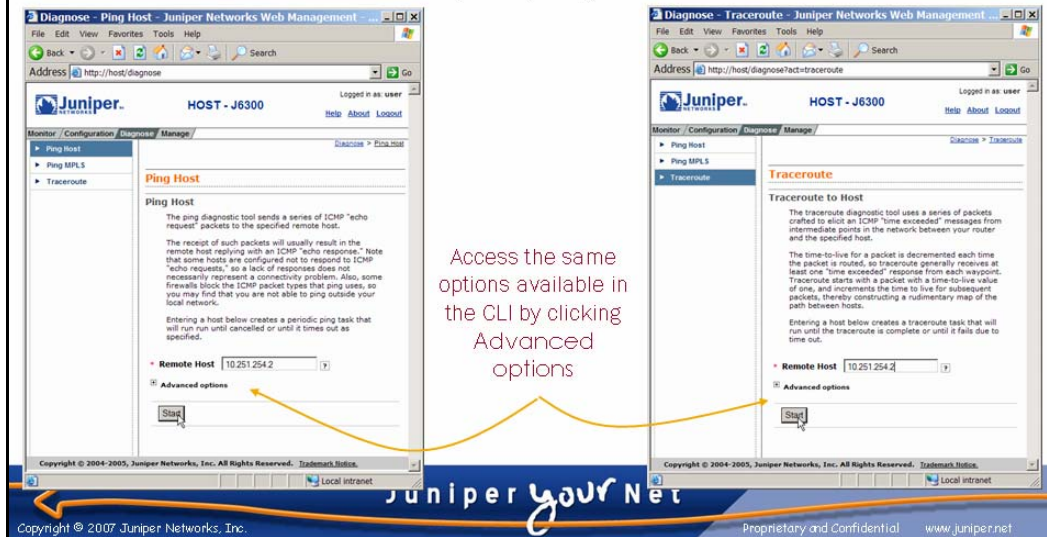


Network Utilities

The slide highlights the topic we discuss next.

Network Utilities: Part 1

- Access ping and traceroute at the Diagnose page
 - Or use the CLI **ping** and **traceroute** commands
 - Use **Ctrl-c** to stop CLI ping and traceroute

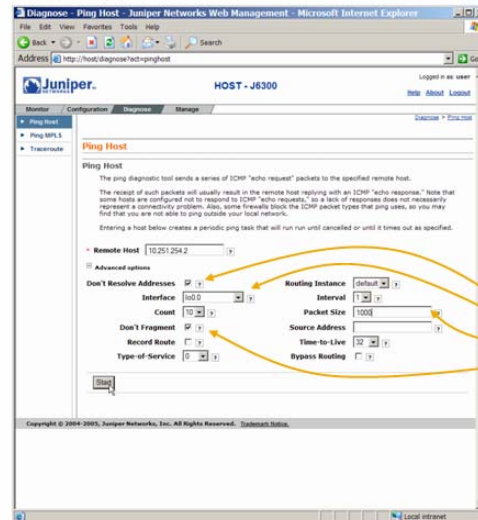


Ping and Traceroute Utilities

The J-Web Diagnose page provides access to the ping and traceroute utilities. You can use these tools to determine general network reachability and the path that packets take to reach a destination. The Advanced options button allows you to specify parameters, such as source IP address and packet size, that can further assist in problem isolation.

J-Web Ping Example

- J-Web ping with Advanced options:

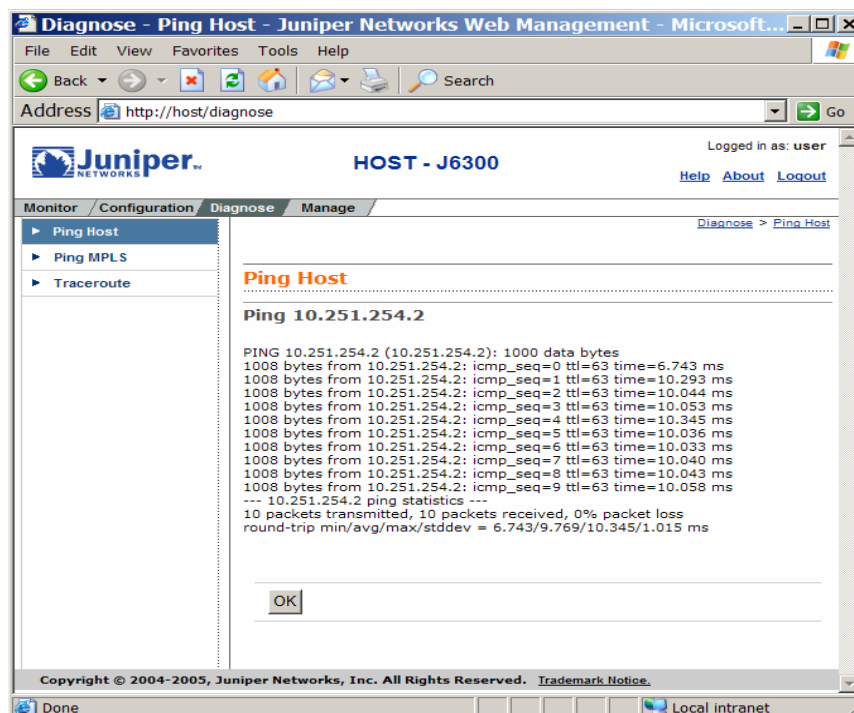


Don't Resolve Addresses,
Interface, Packet Size,
and Don't Fragment



Ping Results

The following capture shows the results of a ping with Advanced options set:



J-Web Traceroute Example

- J-Web traceroute with Advanced options:

The screenshot shows the Juniper J-Web Traceroute interface in a Microsoft Internet Explorer browser window. The address bar shows `http://host/diagnose/fact-traceroute`. The page title is "HOST - J6300". The interface has a navigation bar with "Monitor", "Configuration", "Diagnose", and "Manage". The "Diagnose" tab is selected, and the "Traceroute" sub-tab is active. The "Traceroute to Host" section contains a text description of the tool. Below this, the "Remote Host" field is set to "10.251.254.2". The "Advanced options" section is expanded, showing several settings: "Don't Resolve Addresses" is checked, "Interface" is set to "any", "Time-to-Live" is set to "10", "Type-of-Service" is set to "0", "Resolve AS Numbers" is unchecked, "Routing Instance" is set to "default", "Gateway" is empty, "Source Address" is set to "10.251.254.254", and "Bypass Routing" is unchecked. A yellow callout box with the text "Don't Resolve Addresses, Time-to-Live, and Source Address" has arrows pointing to the "Don't Resolve Addresses" checkbox, the "Time-to-Live" field, and the "Source Address" field. The footer of the page includes the Juniper logo, the text "Juniper your Net", the page number "5-28", and the copyright notice "Copyright © 2007 Juniper Networks, Inc. All Rights Reserved. Trademark Notice.".

Advanced Traceroute Example

This slide demonstrates using traceroute with Advanced options.

Network Utilities: Part 2

- Access Telnet, SSH, and FTP client commands from the CLI

```

user@host> 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
no-resolve Don't attempt to print addresses symbolically
port       Port number or service name on remote host
source     Source address to use in telnet connection

user@host> telnet 127.0.0.1
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.

host (ttypl)

login: user
Password:
. . .

```



Network Utilities

The CLI supports powerful Telnet, SSH, and FTP clients. These clients support various switches that tailor their specific operation.

Network Utilities: Part 3

- The **monitor traffic** command provides CLI access to the tcpdump utility
 - Only displays traffic originating or terminating on local RE
 - The best way to perform analysis of Layer 2 protocols in JUNOS software using **layer2-headers** option
 - Protocol filtering currently requires writing and reading from a file (hidden **write-file** and **read-file** options)

```
user@host> monitor traffic interface tl-4/0/0 layer2-headers
verbose output suppressed, use <detail> or <extensive> for full
protocol decode
Listening on tl-4/0/0, capture size 96 bytes

23:34:22.054988 Out  88 03 IP: 10.251.254.130 > 10.251.254.131: ICMP
echo request seq 15874, length 64
23:34:22.115106 In  10.251.254.131 > 10.251.254.130: ICMP echo reply
seq 15874, length 64
^C
4 packets received by filter
0 packets dropped by kernel
```



Network Utilities and Applications

The **monitor traffic** command provides CLI-based access to the tcpdump utility. This command monitors only traffic originating or terminating on the local RE. This capability is the best way to monitor and diagnose problems at Layer 2 in JUNOS software because tracing, which is similar to debug on other vendors' equipment, does not function for Layer 2 protocols. Tracing is covered on subsequent pages that deal with system logging.

Note that protocol filtering functions (for example, matching on only UDP traffic sent from a specific port) are currently not supported for real-time monitoring. As a workaround, you can write the monitored traffic to a file using the hidden **write-file** option and then read the file with a tcpdump-capable application like Ethereal.

Agenda: Operational Monitoring and Maintenance

- Monitor Platform and Interface Operation
- Network Utilities
- System Logging and Protocol Tracing
- License Management
- Maintaining JUNOS Software
- File System Maintenance and Password Recovery



System Logging and Protocol Tracing

This slide highlights the topic we discuss next.

System Logging and Tracing

- **System logging:**
 - Standard UNIX syslog configuration syntax
 - Primary syslog file is `/var/log/messages`
 - Most processes also write to individual log files
 - Numerous facilities and severity levels are supported
 - The facility defines the class of log message, while the severity level determines the level of logging detail
 - Local and remote syslog support
 - Remote logging (and archiving) recommended for troubleshooting
- **Tracing decodes protocol packets and certain router events**
 - Referred to as *debug* by some other vendors
 - Tracing operations include:
 - Global routing behavior
 - Router interfaces
 - Protocol-specific information



System Logging

System logging (syslog) operations use a UNIX syslog-style mechanism to record system-wide, high-level operations, such as interfaces going up or down or users logging in to or out of the router. You configure these operations by using the **syslog** statement at the `[edit system]` hierarchy level and the **options** statement at the `[edit routing-options]` hierarchy level.

JUNOS software places the results of tracing and logging operations in files that are stored in the `/var/log` directory on the router. You use the **show log file-name** command to display the contents of these files.

Tracing Operations

Tracing operations allow you to monitor the operation of routing protocols by decoding the routing protocol packets that are sent and received. In many ways tracing is synonymous with the debug function on equipment made by other vendors. Note that because of the design of J-series platforms, you can enable reasonably detailed tracing in a production network without negative impact on overall performance or packet forwarding.

Syslog Configuration Example

```
[edit system]
user@host# show syslog
user * {
  any emergency;
}
file messages {
  any notice;
  authorization info;
}
file cli-commands {
  interactive-commands any;
}
file config-changes {
  change-log info;
}
file errors {
  any error;
}

[edit system]
user@host# show syslog | display set
set system syslog user * any emergency
set system syslog file messages any notice
set system syslog file messages authorization info
set system syslog file cli-commands interactive-commands any
set system syslog file config-changes change-log info
set system syslog file errors any error
```

Emergency messages go to all logged-in users

Primary syslog file

Log all CLI commands

Log configuration changes

Log all errors here

The corresponding CLI commands

System Logging Options Example

The slide shows various syslog configuration examples. General syslog configuration options include the following:

- **archive:** Configures how to archive system logging files (default is to keep 10 archive files with a maximum size of 128 K each);
- **console:** Configures the types of syslog messages to log to the system console;
- **facility:** Displays the class of log messages;
- **file filename:** Configures the types of syslog messages to log to the specified file; and
- **files number:** Displays the maximum number of system log files.

You can configure support for explicit priority in syslog messages. This configuration alters the normal syslog message format by adding a numeric priority value. The explicit priority value can simplify the task of parsing log files for important messages. For example, you can search for all messages at priority 7. The presence of explicit priority also accommodates the use of tools that were developed to parse the logs generated by other vendors' equipment.

Interpreting Syslog Messages

- Standard log entries consist of the following fields:
 - Timestamp, platform name, software process name/PID, a message code, and the message text
 - Use **explicit-priority** to include a numeric priority value

```
May 31 23:50:14 host mgd[2711]: %INTERACT-6-
UI_CMDLINE_READ_LINE: User 'user', command 'show version '
```

- Use **help syslog ?** to help interpret message codes

```
user@host> help syslog UI_CMDLINE_READ_LINE
Name:          UI_CMDLINE_READ_LINE
Message:       User '<username>', command '<input>'
Help:          User entered command at CLI prompt
Description:   The indicated user typed the indicated command at the CLI prompt
                and pressed the Enter key, sending the command string to the
                management process (mgd).
Type:          Event: This message reports an event, not an error
Severity:      info
```



Interpreting System Log Entries

When using the standard syslog format, each log entry written to the messages file consists of the following fields:

- timestamp: Indicates when the message was logged.
- name: Displays the configured system name.
- Process name/PID: Displays the name of the process (or the process ID when a name is not available) that generated the log entry.
- message-code: Provides a code that identifies the general nature and purpose of the message. In the example shown, the message code is UI_CMDLINE_READ_LINE.
- message-text: Provides additional information related to the message code.

When the **explicit-priority** statement is added, the syslog message format is altered to include a numeric priority value. In this case the value 0 is used for the most significant and urgent messages (emergency), while 7 is used to denote debug-level messages.

Continued on next page.

Interpreting Message Codes

Consult the *System Log Messages Reference* documentation for a full description of the various message codes and their meanings. Or, better yet, use the CLI's `help` function to obtain this information. The example shows the operator obtaining help on the meaning of the `UI_CMDLINE_READ_LINE` message code. Based on the output, it becomes relatively clear that the message code shows a command that a user entered at the CLI prompt.

Tracing Overview

- Tracing is the JUNOS software equivalent of *debug*
 - Can be enabled on a production network
 - Requires configuration
 - Multiple options (flags) can be traced to a single file
- Generic tracing configuration syntax:

```
[edit protocols protocol-name]
user@host# show
  traceoptions {
    file filename [replace] [size size] [files number] [no-
stamp];
    flag flag [flag-modifier] [disable];
  }
```

The protocol/function being traced

Where to write the trace results

Flags identify what aspects of the protocol is traced and at what level of detail



Hear Tracing, Think Debug

Tracing is the JUNOS software term for what other vendors sometimes call *debug*. In most cases when you enable tracing (through configuration), you create a trace file that is used to store decoded protocol information. You analyze these files using standard CLI log file syntax like **show log logfile-name**. Because of the design of Juniper Networks routing platforms, you can enable detailed tracing in a production network without significantly impacting performance. Even so, you should always remember to turn tracing off once you have completed your testing to avoid unnecessary resource consumption.

Generic Tracing Configuration

The slide shows a generic tracing stanza, which, if applied to the [edit routing-options] portion of the configuration hierarchy, would result in global tracing of routing events. Global routing protocol tracing operations track all general routing operations and record them in the specified log file. The individual routing protocols inherit any global tracing operations that you configure. To modify the global tracing operations for an individual protocol, configure tracing when configuring that protocol.

Continued on next page.

Generic Tracing Configuration (contd.)

Configuration options for tracing are the following:

- **file filename**: Specifies the name of the file in which to store information.
- **size size**: Specifies the maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB). When a trace file named *trace-file* reaches this size, it is renamed *trace-file.0*. When the trace file again reaches its maximum size, *trace-file.0* is renamed *trace-file.1*, and *trace-file* is renamed *trace-file.0*. This renaming scheme continues until the maximum number of trace files is reached. Then the oldest trace file is overwritten. If you specify a maximum file size, you also must specify a maximum number of trace files with the **files** option. The default size is 1 MB.
- **files number**: Specifies the maximum number of trace files. When a trace file named *trace-file* reaches its maximum size, it is renamed *trace-file.0*, then *trace-file.1*, and so on, until the maximum number of trace files is reached. Then the oldest trace file is overwritten. The default is ten files.
- **no-stamp**: Prevents timestamp information from being placed at the beginning of each line in the trace file. By default, if you omit this option, timestamp information is placed at the beginning of each line of the tracing output.
- **replace**: Replaces an existing trace file if one exists. By default, if you omit this option, tracing output is appended to an existing trace file.

Including the **traceoptions** statement at the `[edit interfaces interface-name]` hierarchy level allows you to trace the operations of individual router interfaces. You can also trace the operations of the interface process, which is the device-control process (*dcd*).

When tracing a specific interface, the specification of a trace file is not supported. The JUNOS software kernel does the logging in this case, so the tracing information is placed in the system's *messages* file. In contrast, global interface tracing supports an archive file; by default `/var/log/dcd` is used for global interface tracing.

Protocol Tracing (1 of 2)

- Include the **traceoptions** statement at the `[edit protocols protocol-name]` hierarchy
 - Useful when troubleshooting configuration and interoperability problems



Protocol Tracing

You trace the operations of a specific protocol by including the **traceoptions** statement at the `[edit protocols protocol-name]` hierarchy. In most cases you will want to be a bit selective in what you trace because selecting the **all** keyword will likely numb your mind with endless minutia. The sample BGP stanza on the slide reflects a typical tracing configuration that will provide details about important events like open messages or BGP route updates. In most cases you will want to use the **detail** switch to a given protocol flag for the added information often needed in troubleshooting scenarios.

Protocol Tracing (2 of 2)

- A typical BGP tracing configuration is shown along with sample output:

```
[edit protocols bgp]
user@host# show
traceoptions {
    file bgp-trace;
    flag open detail;
    flag update detail;
    flag keepalive detail;
}
user@host# run show log bgp-trace
. . .
Feb 19 16:07:47 BGP RECV 192.168.2.1+2705 -> 192.168.0.1+179
Feb 19 16:07:47 BGP RECV message type 1 (Open) length 45
Feb 19 16:07:47 BGP RECV version 4 as 10 holdtime 90 id 192.168.2.1
    parmlen 16
Feb 19 16:07:47 BGP RECV MP capability AFI=1, SAFI=1
. . .
```



Sample Output

A sampling of the results obtained with the tracing configuration are shown. As with any log file, you simply enter a **show log trace-file-name** command to view the decoded protocol entries. The sample trace output reflects the receipt of a BGP open message from 192.168.2.1 and goes on to show some of the session parameters that are being proposed and which must be agreed upon for successful BGP session establishment (for example, hold-time and address family support).

Analyzing Log and Trace Files

- Use the `show log file-name` CLI command to display contents of log and tracefiles
 - Hint: Get help on available options at the `more` prompt by entering an `h`
- Do not forget the CLI's pipe functionality; it makes log parsing a breeze!

```
user@host> show log messages | match "support info"
May 31 23:49:16 host mgd[2711]: %INTERACT-6-UI_CMDLINE_READ_LINE:
User 'user', command 'request support information'
May 31 23:49:24 host mgd[2711]: %INTERACT-6-UI_CMDLINE_READ_LINE:
User 'user', command 'request support information | no-more'
```

- Cascade pipe instances to evoke a logical AND search; use quotes to evoke a logical OR, as shown:

```
user@host> show log messages | match "error | kernel | panic"
```



Viewing Logs and Traces

By default, log and trace files are stored in `/var/log`. To view stored log files, use the command `show log`. Recall that the CLI automatically pauses when there is more than one screen's worth of information, and that at this `more` prompt, you can enter a forward slash (`/`) character to conduct a forward search. As a hint, enter `h` when at a `more` prompt for the context help screen of available commands:

```
---(Help for CLI automore)---
Clear all match and except strings:          c or C
Display all line matching a regexp:          m or M <string>
Display all lines except those matching a regexp: e or E <string>
Display this help text:                     h
Don't hold in automore at bottom of output: N
Hold in automore at bottom of output:       H
Move down half display:                    TAB, d, or ^D
Move down one line:                        Enter, j, ^N, ^X, ^Z, or Down-Arrow
. . .
```

Being able to cascade multiple instances of the CLI's pipe functionality is a real benefit when you must search a long file for associated entries. You can also search for multiple criteria in a logical OR fashion as shown by the example that searches for lines that include any of the words `error`, `kernel`, or `panic`.

Miscellaneous Log File Commands

- Monitor a log or a trace in real time with the CLI's **monitor** command

```
user@host> monitor start filename
```

- Shows updates to monitored file(s) until canceled, with piped output matching!
- Use **Esc-q** to halt and resume real-time output to screen
- Issue **monitor stop** to cease all monitoring

- Log and trace file manipulation:

- Use the **clear** command to truncate (clear) log and trace files

```
user@host> clear log filename
```

- Use the **file delete** command to delete log and trace files

```
user@host> file delete filename
```



Monitoring Logs and Trace Files

Use the **monitor start** CLI command to view real-time log information. You can monitor several log files at one time. The messages from each log are identified by filename, where filename is the name of the file from which entries are being displayed. This line is displayed initially and when the CLI switches between log files.

Using **Esc-q** enables and disables syslog output to the screen; using **monitor stop** ceases all monitoring. Note that you can use the CLI's **match** functionality to monitor a file in real time, while only displaying entries that match your search criteria. To make use of the functionality, use a command in the form of:

```
user@host> monitor start messages | match fail
```

If you do not delete or disable all trace flags, tracing continues in the background, and the output continues to be written to the specified file. The file remains on the RE's compact flash drive until either it is deleted manually or overwritten according to the **traceoptions** file parameters. To disable all tracing at a particular hierarchy, issue a **delete traceoptions** command at that hierarchy, and commit the changes.

Log and Trace File Manipulation

To truncate files used for logging, use the **clear log filename** command.

To delete a file, use the **file delete** command. If you want, you can also use wildcards with the file command's **delete**, **compare**, **copy**, **list**, and **rename** operations.

Agenda: Operational Monitoring and Maintenance

- Monitor Platform and Interface Operation
- Network Utilities
- System Logging and Protocol Tracing
- License Management
- Maintaining JUNOS Software
- File System Maintenance and Password Recovery



License Management

The slide highlights the topic we discuss next.

J-series License Overview

- Various software features require licensing
 - Currently uses a soft model that does not disable unlicensed functions
 - A lack of license results in commit warnings and log entries
- Device and group licenses:
 - A device license is tied to a specific device ID stored in EPROM and can only be used on that device
 - A device license can define one or more groups
 - A group license is tied to a group of devices
 - Corresponding group definition must be present
- Licenses are not part of the configuration
 - Stored locally on the device, possibly from the factory
 - Opaque data to JUNOS software
 - Licenses can exist for unsupported features; license is only referenced when a feature requires it



Some Features Require Licensing

Various software features require licensing on J-series units. The soft licensing model ensures that a licensing problem will never be the cause of a network outage.

Device and Group Licenses

A device license authorizes you to configure certain software features on the J-series router, or associates a router with a group license. However, because group licenses are currently unsupported, the group information associated with licenses will always be blank. A device license that ties a specific device to a group membership must be present to activate the corresponding group license's software features and hardware ports. Multiple device licenses, group licenses, or both in multiple license files can be present. Overlapping software feature licenses do not create a conflict.

Not Part of the Configuration

Note that license keys are not stored as part of the device's configuration, but as individual files in the compact flash drive's `/config/license` directory. Divorcing the license data from the configuration makes unintentional deletion or modifications extremely unlikely.

J-series License Requirements

- Software features:
 - Traffic Analysis
 - Advanced BGP
 - IBM Networking
- No licenses for hardware are required



Software Feature Licenses

The following list shows the various licenses and the hierarchies to which they relate. Any statements configured under the listed hierarchies require the related license:

- *Traffic Analysis / JFlow*: [edit forwarding-options sampling] and [edit forwarding-options accounting]
- *Advanced BGP / BGP route reflectors*: [edit protocols bgp cluster]
- *IBM Networking / data-link switching (DLSw)*: [edit protocols dlsw]

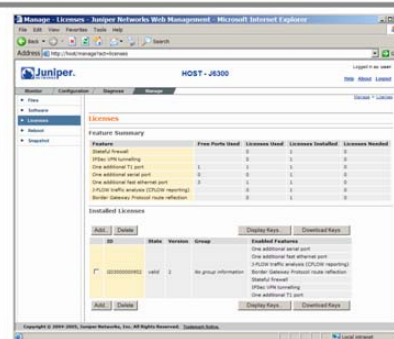
Hardware Port Licenses

Licensing requirements for J-series hardware ports were removed beginning with JUNOS software Release 7.6.

Obtaining a J-series License

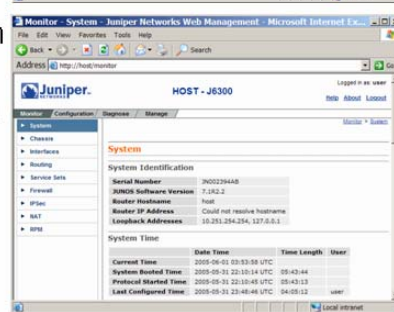
■ Preinstalled:

- No configuration is required
- Verify from J-Web Manage > Licenses page or with **show system license keys** CLI command



■ Authorization code:

- Provided on a piece of paper with your shipment
- Obtain router serial number from J-Web Monitor > System page or with **show chassis hardware** CLI command
- Go to http://www.juniper.net/generate_license



Preinstalled Licenses

Your J-series router might arrive with the license file preinstalled. No further configuration is required to use the license, but you should back up your license data using the J-Web Download Keys... button on the J-Web Manage > Licenses page or the **request system license save** CLI command.

Generate a License from an Authorization Code

Alternatively, you might receive a piece of paper with your J-series router that provides an authorization code. You can then use this authorization code to generate a license key using the tool at http://www.juniper.net/generate_license. You will also need your router's serial number to bind the newly generated license to a particular router. Obtain this number with the **show chassis hardware** CLI command or from J-Web's Monitor > System page. Note that the license file generated by the http://www.juniper.net/generate_license tool might include comments in addition to the license data. You must remove these comments before installing the license file on the router.

Managing Licenses: CLI

■ Manage license keys using the CLI

- Display license usage:
`show system license usage`
- Display license key:
`show system license keys`
- Load, save, or delete licenses:
`request system license [add | delete | save]`

lab@London> **show system license usage**

Feature name	Free ports used	Licenses used	Licenses installed	Licenses needed
firewall		0	1	0
ipsec-vpn		0	1	0
if-se	1	0	1	0
if-fe	3	0	1	0
j-flow		0	1	0
sla		0	1	0
bgp-reflection		0	1	0

2 built-in ports plus
2 PIM ports are
used

No licenses are
actually used

Several features and two additional
interface licenses are installed

No additional licensing
needed

Managing Licenses at the CLI

The CLI allows you to display licenses installed by using the **show system license keys** command, while the **show system license usage** command shows which of the installed licenses are being used and any additional licenses that are required.

You can add licenses from, or save licenses to, a local file on the router, a remote URL, or the terminal.

Installed licenses are stored in the router's `/config/license/` directory as license-key.lic. There should be no need to directly alter these files. Use J-Web or the **request system license** CLI command instead.

Managing Licenses: J-Web

- Access J-Web license management at the Manage > Licenses page

The screenshot shows the Juniper J-Web interface for a router named 'LONDON - J4300'. The user is logged in as 'root'. The 'Manage' tab is selected, and the 'Licenses' page is displayed. The page includes a sidebar with navigation options like Files, Software, Licenses, Reboot, and Snapshot. The main content area shows a 'Feature Summary' table and an 'Installed Licenses' table. Annotations with yellow arrows point to the 'Add' button in the 'Installed Licenses' table and the 'Download Keys' button, with text explaining that license keys can be added via terminal paste or URL and should be downloaded for safe keeping.

Feature	Free Ports Used	Licenses Used	Licenses Installed	Licenses Needed
Stateful firewall	0	0	1	0
IPSec VPN tunneling	0	0	1	0
One additional serial port	1	0	1	0
One additional fast ethernet port	2	0	1	0
J-Flow traffic analysis (CFLW reporting)	0	0	1	0
Service Level Agreement monitoring	0	0	1	0
Border Gateway Protocol route reflection	0	0	1	0

ID	State	Version	Group	Enabled Features
G03000000593	valid	2	No group information	One additional serial port One additional fast ethernet port J-Flow traffic analysis (CFLW reporting) Service Level Agreement monitoring Border Gateway Protocol route reflection Stateful firewall IPSec VPN tunneling

Managing Licenses with J-Web

J-Web also offers a GUI equivalent to the **request system license** and **show system license** commands. You can find this equivalent at the Manage > Licenses page.

Agenda: Operational Monitoring and Maintenance

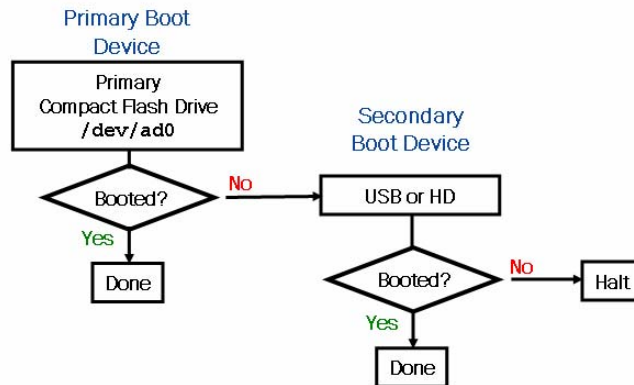
- Monitor Platform Operation
- Network Utilities
- System Logging and Protocol Tracing
- License Management
- ➔ **Maintaining JUNOS Software**
- File System Maintenance and Password Recovery



Maintaining JUNOS Software

The slide highlights the topic we discuss next.

The Default Boot Sequence



- The compact flash drive is the primary boot device
 - On J-series platforms, USB is the secondary device
 - On M-series platforms, the hard drive is the secondary device

Compact Flash Drive Is Primary Boot Device

The software is installed on the router's primary compact flash drive (a nonrotating drive). If there is a problem with the software installed on the primary flash drive, the router attempts to boot from alternative devices when possible. All J-series platforms support the use of front-panel USB flash memory as a boot device. M-series platforms have a hard drive as a secondary boot device, and a PCMCIA slot or USB port that can be used in emergencies.

Choosing a Boot Device

- Boot device can be specified:
 - Using Reboot From Media option on J-Web Manage > Reboot page
 - When requesting a system reboot from the CLI:

```
user@host> request system reboot media ?
Possible completions:
compact-flash      Standard boot off flash device
usb               Boot off USB device
```

- Warnings issued at CLI login when booted from alternative device

```
host (ttyd0)

login: user
Password:

--- JUNOS 8.1R2.4 built 2006-12-29 08:27:34 UTC
---
--- NOTICE: System is running on alternate media device (/dev/da0s1a).
---
```



Specifying the Boot Device

Selecting a boot device is desirable for some maintenance actions. You can reboot the router from a specific device using the Reboot From Media option on the J-Web Manage > Reboot page or using the media option to the **request system reboot** CLI command.

Booting from an Alternative Device

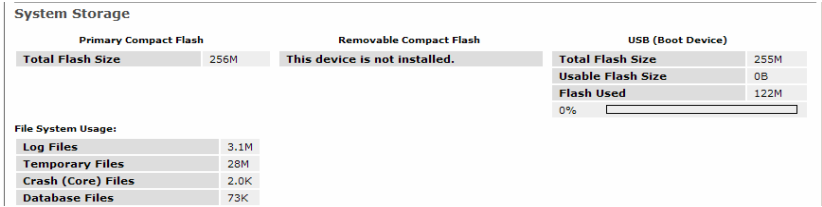
When the router is booted from a device other than the primary boot device, you receive a warning message upon CLI login. The message states that the router booted from an alternative device and gives the device name. The following device names are used:

- /dev/ad0s1a: Primary compact flash drive;
- /dev/ad1s1a: M-series hard drive;
- /dev/ad2s1a: J4300/J6300 removable compact flash drive; and
- /dev/da0s1a: USB storage device.

Continued on next page.

Specifying the Boot Device (contd.)

The System Storage section of the J-Web Monitor > System page also shows from which device the router is booted. This screen capture shows a router booted from USB:



J-series Software Packaging

- **Software packaging:**
 - J-series software packages come in domestic and export versions
 - Domestic version supports 128-bit encryption and requires signed agreement
 - Export version supports only 56-bit encryption
 - Packages are signed using the Secure Hash Algorithm 1 (SHA-1) to ensure integrity
- **JUNOS software will only execute signed binaries**
- **No individual software component upgrades, install packages, or removable media packages**
 - The snapshot function is used to place a bootable image on new or backup boot device
 - J-series packaging forces the upgrade or downgrade of all software components



J-series Software Packaging

While the J-series JUNOS software is built from the same code base as M-series and T-series software, it is packaged differently. You can install a J-series JUNOS software package only on a J-series router. There are two available packages for each J-series JUNOS software version. The domestic package includes 128-bit encryption and is subject to United States encryption export restrictions, while the export version includes only 56-bit encryption and is available worldwide.

Unlike the M-series and T-series JUNOS software packages, the J-series export package does include SSH support with single Data Encryption Standard (DES) encryption only. This can be confusing because few SSH clients still support this weak encryption. The result is that you can configure the SSH service with the export package, but most SSH clients will be unable to connect.

Signed Binaries

Juniper Networks routers run binaries supplied by Juniper Networks only. Each JUNOS software image includes a digitally signed manifest of executables, which are registered with the system only if the signature can be validated. JUNOS software does not execute any binary without a registered fingerprint. This feature is designed to protect the system against unauthorized software and activity that might compromise the integrity of your router.

Continued on next page.

No Component Upgrades

The JUNOS software package is comprised of several different components, and you can see these components listed with the **show version detail** CLI command. You cannot, however, upgrade these components individually. J-series software packaging includes all components and forces the upgrade or downgrade of all software components.

J-series Package Naming Convention

- J-series software packages are named as follows:

`junos-jseries-m.nZnumber-region.tgz`

- m.n is the major version number
- Z is a single uppercase letter
 - A: Alpha
 - B: Beta
 - R: Release
 - I: Internal
- number is the release number; might include the build number for that release
- region is either domestic or export
- Example: `junos-jseries-7.1R2.2-domestic.tgz`



Package Naming

A JUNOS software package has a name in the format

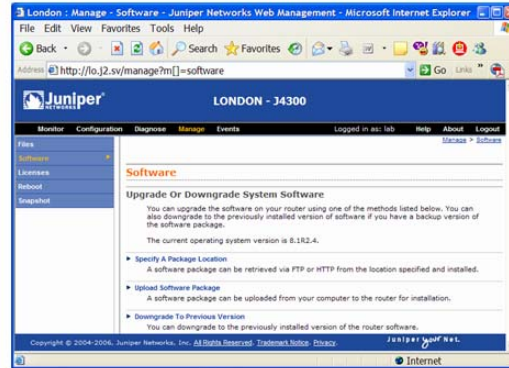
`package-m.nZnumber-region.tgz`:

- package is a description of the software contents. This description is `junos-jseries` for all J-series software images.
- m.n are two integers that represent the software release number.
- Z is a capital letter that indicates the type of software release. In most cases, it is an R to indicate that this is released software. If you are involved in testing prereleased software, this letter might be an A (for alpha-level software), B (for beta-level software), or I (for internal, test, or experimental versions of software).
- A number represents the version of the software release and includes the internal build number for that version. For example, `junos-jseries-7.1R2.2-domestic.tgz` indicates a JUNOS software bundle associated with version 7.1, release 2, build 2.
- region will be either `domestic` or `export`. Domestic versions include strong encryption, while export versions do not.

Again, ensure that you always load J-series bundles on J-series platforms only.

Upgrading JUNOS Software

- Download and install a new package
 - Use J-Web Manage > Software page to download and install a package from a remote server or the local PC
 - Or use the **request system software add** CLI command
 - Locally stored packages should be kept in `/var/tmp` for easy cleanup
 - Look out for problems relating to low storage space
 - File system clean up is covered in a subsequent section



Installing Software

You can upgrade the JUNOS software from either J-Web or the CLI. You should store JUNOS bundles in the router's `/var/tmp` directory. You can easily clean up files stored in this directly using the J-Web file cleanup wizard.

The default primary compact flash drive sizes do not have enough space to store several JUNOS software versions. Always check available storage capacity before downloading a new JUNOS bundle. You can view available capacity from the System Storage section of the J-Web Monitor > System page or with the **show system storage** CLI command:

System Storage

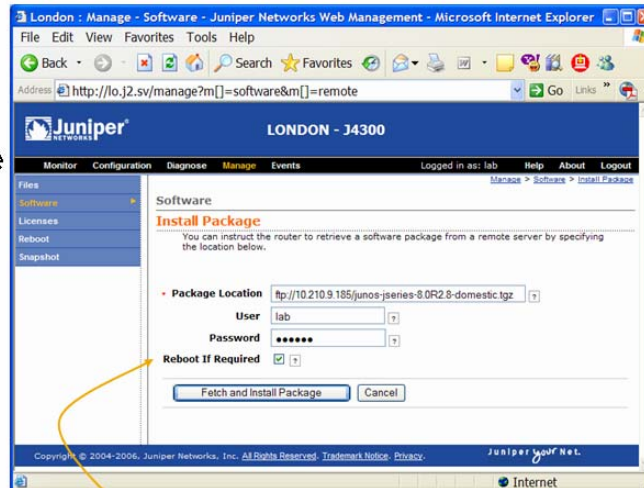
Primary Compact Flash (Boot Device)	
Total Flash Size	256M
Usable Flash Size	131M
Flash Used	122M
93.12% <div style="width: 93.12%; height: 10px; background-color: red; border: 1px solid black;"></div>	

```
user@host> show system storage
```

Filesystem	Size	Used	Avail	Capacity	Mounted on
/dev/ad0s1a	131M	121M	7.7M	94%	/
...					

Upgrade Example (1 of 2)

- Use J-Web
Manage >
Software >
Install Package
page to install a
package from a
remote server
 - An FTP-based
URL is shown
in this example



A reboot is required to activate new software

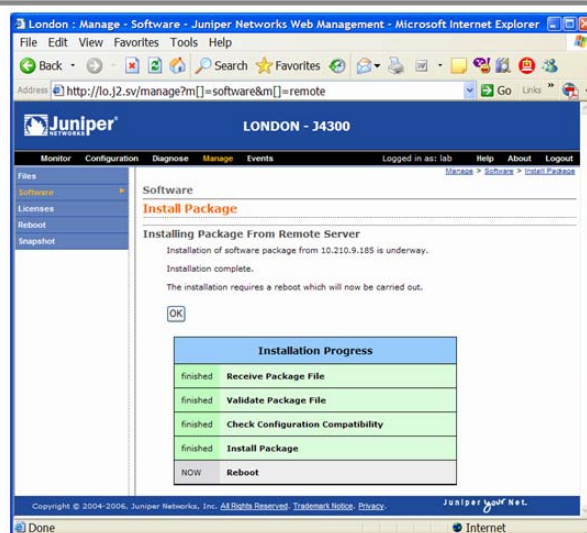
Installing Software from a Remote Server with J-Web

Use the J-Web Manage > Software > Install Package page to specify a remote URL that contains a JUNOS software bundle. The software will be downloaded and installed. To activate the new software you must reboot the router. You can perform this reboot directly from the Manage > Software > Install Package page using the Reboot If Required check box, or you can reboot later using the Manage > Reboot page. You can also install software using the **request system software add** CLI command.

You can also use the J-Web Manage > Software > Upload Package page to copy software directly from your PC to the router.

Upgrade Example (2 of 2)

- You are presented with status indications as the upgrade process proceeds
 - Watch for any error messages during the upgrade



J-Web Software Upgrade Status

As the JUNOS software is being installed from the remote location, you will be presented with an autoupdating page that shows the current status. Watch for any error messages indicating a problem with the upgrade.

Rollback System Software

- By default, a copy of the previous software package is stored in `/packages/`
 - Use the **request system software rollback** CLI operational-mode command or the J-Web Manage > Software > Downgrade page to restore the previous version
 - Normally used when problems are encountered with new software

```
user@host> request system software rollback
junos-7.1R1.3-domestic 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
```

```
user@host> request system reboot
Reboot the system ? [yes,no] (no) yes
```

- Backup package can be deleted through J-Web or CLI to conserve space as needed



Rollback to Previous Software

When you upgrade software the previous JUNOS software bundle is saved in the `/packages/` directory. If you encounter problems with the new software, you can quickly revert to the previous version using the J-Web Manage > Software > Downgrade page or the **request system software rollback** CLI command.

If you become low on storage space, you can delete the previous software package using the **request system software delete-backup** CLI command or the J-Web Manage > Files > Delete backup JUNOS package link.

Backing Up Existing Software



- Back up system software and configuration to alternative boot device:
 - Before major upgrade to ensure system recovery if necessary
 - After upgrade when system is judged to be stable; permits recovery from primary compact flash drive failures
- Performed with the CLI **request system snapshot** command or the J-Web Manage > Snapshot page

Backup Options

In the event of a failure on the primary compact flash drive, the router can boot from an alternative device in the form of removable compact flash drive or a USB storage device. It is possible to have one version of JUNOS software on the primary compact flash drive and another version of JUNOS software on an alternative device. But what if you want to ensure that the primary compact flash drive and alternative device versions of JUNOS software are exactly the same?

Requesting a System Snapshot

When the router is booted from the primary compact flash drive, you can use the J-Web Manage > Snapshot page or the CLI **request system snapshot** command to mirror the contents of the primary compact flash drive onto a secondary boot device. When the router is booted from an alternative device, a snapshot mirrors the environment on the alternative device to the router's primary compact flash drive, by default. You can also specify the target device as an argument to the snapshot command.

You should back up software before you upgrade JUNOS software. Or, after you upgrade the software on the router and are satisfied that the new packages are successfully installed and running, you should consider issuing a snapshot to back up the software onto an alternative device.

Continued on next page.

Requesting a System Snapshot (contd.)

In general, system snapshots are best used to preserve a known good environment when performing upgrades or downgrades on the router's flash memory. In these cases, having the previous environment backed up on the alternative device allows you to return the router to its previous state if the flash-based upgrade or downgrade should fail or exhibit operational problems.

Be sure that you do not remove a storage device when it is in use. Alternative storage devices might be in use because a snapshot is underway, the router has booted from the alternative device, or a crash file is being written. Once the snapshot process is completed, you can remove the USB storage device for safe keeping, or leave the alternative device inserted to recover from primary compact flash drive failures (which require a reboot).

Snapshot Options

- Snapshot supports several important arguments:
 - **media**: Explicitly identify the target device; source device is always the current boot device
 - **as-primary**: Prepares the removable compact flash drive for use in the primary compact flash slot
 - Needed so that the root partition can be mounted correctly when used in the primary compact flash drive slot
 - Not applicable to USB device
 - Not necessary with software Release 7.5 and later
 - **partition**: Lays down a new partition table; required when using new (non-JUNOS software formatted) device
 - **factory**: Only copy factory-default files

```
user@host> request system snapshot media ?
Possible completions:
compact-flash      Write snapshot to compact flash
removable-compact-flash Write snapshot to removable compact flash
usb                Write snapshot to device connected to USB port
```



Snapshot Options

The **request system snapshot** CLI command and J-Web Manage > Snapshot page support several options. The **media** (Target Media in J-Web) option specifies the destination of the snapshot. The source device is always the current boot device.

The **as-primary** option is only used if you are performing a snapshot to a compact flash device that is currently, or will be, installed in the primary compact flash drive slot. If you are performing a snapshot to the removable compact flash device of a J4300 or J6300 router running a JUNOS software version prior to Release 7.5, you must use the **as-primary** option if you will later install this compact flash device in the primary compact flash slot of a J-series router. Otherwise, you do not need to use the **as-primary** option.

The **partition** switch forces the device to be reformatted with the JUNOS partitioning scheme. This switch is required when using a new (non-JUNOS software-formatted) device, but you can use it even if the device was previously formatted with the JUNOS partitioning scheme.

The **factory** switch only copies factory-default files to the alternative device. Thus, your current and rollback configurations will not be backed up to the alternative device.

Snapshot Example: New Compact Flash Device

- The **partition** switch is needed to prepare a new device for a snapshot
 - Copies from the current boot device to the other device

```

user@host>
Removable Compact Flash inserted
ata2 at port 0x170-0x177,0x376 irq 15 on isa0
ad2: 244MB <Hitachi XXM2.3.0> [695/15/48] at ata2-master using BIOSPIO

user@host> request system snapshot partition
Clearing current label...
Partitioning ad2 ...
Running newfs (134Mb) on ad2s1a...
Running newfs (24Mb) on ad2s1e...
Copying '/dev/ad0s1a' to '/dev/ad2s1a' .. (this may take a few minutes)
Copying '/dev/ad0s1e' to '/dev/ad2s1e' .. (this may take a few minutes)
The following filesystems were archived: / /config

user@host>
    
```

Backup compact flash device is detected upon insertion

The partition switch results in the creation of a new partition table

The contents of the root (/) and /config file systems are copied to target device



Partitioning New Devices

This slide demonstrates using the **partition** switch to format a new compact flash device.

Agenda: Operational Monitoring and Maintenance

- Monitor Platform and Interface Operation
- Network Utilities
- System Logging and Protocol Tracing
- License Management
- Maintaining JUNOS Software
- ➔ File System Maintenance and Password Recovery



File System Maintenance and Password Recovery

The slide highlights the topic we discuss next.

J-series File System Overview

- Key directory and file locations include:
 - `/`: The root file system—housed on boot device
 - `/config`: Location for the active configuration (`juniper.conf.gz`), first 3 rollbacks, rescue configuration, and license data—housed on boot device
 - `/var`: User homes, log file, and temporary storage
 - `/var/db/config`: Location of rollback indexes 4–49
 - `/var/home`: Nonroot user home directories
 - `/var/log`: Location of system log (and trace) files
 - `/var/tmp`: Location of various temporary files, such as core dumps, and the recommended storage area for JUNOS software packages
 - `/var/sw/pkg/`: Storage for packages successfully installed for use with software rollback operations



Overview of the JUNOS Software File System

The following list shows the key directories and file locations:

- `/`: The root file system. Located on the router's boot device, which is normally the primary compact flash drive.
- `/config`: This directory is located on the boot device and contains the current operational router configuration and the last three committed configurations as well as the rescue configuration if one is saved. The `/config/license` directory holds any license files that are loaded.

Continued on next page.

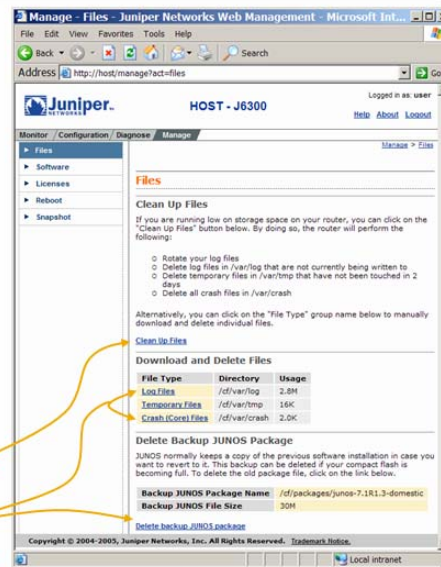
Overview of the JUNOS Software File System (contd.)

- `/var`: This directory is also located on the boot device. This file system contains the following subdirectories:
 - `/var/db/config`: Up to 46 additional previous versions of committed configurations, which are stored in the files `juniper.conf.4.gz` through `juniper.conf.49.gz`.
 - `/var/home`: Contains users' home directories, which are created when you create user access accounts. For users using SSH authentication, their `.ssh` file, which contains their SSH key, is placed in their home directory. When users save or load configuration files, those files are loaded from their home directory unless the users specify a full path name.
 - `/var/log`: Contains system log and tracing files.
 - `/var/tmp`: Contains daemon core files (if present) and a copy of the last software package installed.
 - `/var/sw/pkg/rollback`: The `rollback` file identifies the previous software version for use with software rollback operations. The matching package file should be located in the `/packages` directory, assuming that one exists and that the rollback file was not deleted to save compact flash space.

File System Cleanup

- Compact flash space is limited—view usage with J-Web Monitor > System page
- Use the J-Web Manage > Files page to free up space
 - Often required to complete software upgrades
- Or use CLI **file delete** and/or **request system software delete-backup** commands

Cleanup wizard
Manual cleanup



Limited Space

The compact flash drive used for primary storage on J-series routers is somewhat limited in comparison to the hard drives found on M-series and T-series routers. While the router will continue to forward traffic if the compact flash drive becomes full, on a J-series router, you will lose log messages and be unable to modify the configuration until space is freed. You can monitor usage in the System Storage section of the J-Web Monitor > System page or by using the **show system storage** CLI command.

File System Cleanup

In most cases you can simply use the J-Web file system cleanup wizard found on the Clean Up Files link of the J-Web Manage > Files page to locate and remove files that are no longer necessary. You can also manually remove files using the other links on the Manage > Files page or with the CLI **file delete** command, the **request system software delete-backup** command, or both. The file system cleanup wizard identifies only unneeded JUNOS software packages in the /var/tmp directory. Avoid storing JUNOS software packages in other location.

Continued on next page.

Freeing Space from the CLI

You can also manually free storage space from the CLI. Use the **file delete file-name** command to remove unnecessary files. The **request system software delete-backup** command removes the backup JUNOS package, if present.

Password Recovery Process

Steps:

1. Obtain console access and reboot the system
 - At the boot loader prompt, enter a space character to obtain the `ok` prompt
 - Enter `boot -s` to boot into single-user mode
 - When prompted, enter `recovery`

System watchdog timer disabled

Enter full pathname of shell or `'recovery'` for root password recovery
or RETURN for `/bin/sh`: `recovery`

2. The system performs disk checks and places you at the `root@host>` CLI prompt
3. Follow on-screen instructions to enter configuration mode and reset root password
4. Commit the changes and reboot the system



Password Recovery

Password recovery requires several steps to complete. To recover a lost root password, perform these steps:

1. Obtain console access and reboot the system. Watch as the system boots, and enter a space character at the boot loader quick help menu to get a command prompt. Enter `boot -s` at the prompt to boot into single-user mode as shown:

```
FreeBSD/i386 bootstrap loader, Revision 0.8
(builder@harrekki.juniper.net, Wed May  5 09:12:39 GMT 2004)
Loading /boot/defaults/loader.conf
/kernel text=0x49e827 data=0x2f3ac+0x49c48 syms=[0x4+0x40960+0x4+0x4d68a]
```

```
Hit [Enter] to boot immediately, or space bar for command prompt.
Booting [kernel] in 8 seconds...
```

```
<user enters space>
```

```
Type '?' for a list of commands, 'help' for more detailed help.
```

```
ok boot -s
```

```
Copyright (c) 1996-2001, Juniper Networks, Inc.
```

```
All rights reserved.
```

```
Copyright (c) 1992-2001 The FreeBSD Project.
```

```
Copyright (c) 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, 1994
```

```
The Regents of the University of California. All rights reserved.
```

```
. . .
```

Continued on next page.

Password Recovery (contd.)

2. The system performs a single-user boot-up process and prompts the user to run the recovery script or enter the path to a shell. Enter **recovery** to run the password recovery script:

```
. . .
Mounted junos package on /dev/vn0...
System watchdog timer disabled
Enter full pathname of shell or 'recovery' for root password recovery or RETURN
for /bin/sh: recovery
. . .
```

3. The system finishes booting and places you at a **root>** CLI prompt. Follow the on-screen steps to enter configuration mode and edit or delete the root password. During this time you *might* see some error messages regarding the configuration database version. These messages are normal, and they should not impact your ability to complete the password recovery process:

```
. . .
Performing checkout of management services ...

NOTE: Once in the CLI, you will need to enter configuration mode using
NOTE: the 'configure' command to make any required changes. For example,
NOTE: to reset the root password, type:
NOTE:     configure
NOTE:     set system root-authentication plain-text-password
NOTE:     (enter the new password when asked)
NOTE:     commit
NOTE:     exit
NOTE:     exit
NOTE: When you exit the CLI, you will be asked if you want to reboot
NOTE: the system
```

```
Starting CLI ...
root@host>
```

4. After changing the password, commit the change and exist the CLI. Enter **y** at the prompt to reboot the system:

```
[edit]
root@host# commit and-quit
Exiting configuration mode

root@host> exit

Reboot the system? [y/n] y
Waiting (max 60 seconds) for system process `bufdaemon' to stop...stopped
Waiting (max 60 seconds) for system process `syncer' to stop...stopped

syncing disks...
```

Review Questions

1. List two methods for monitoring J-series platform operation.
2. What does a blinking power LED indicate?
3. What command displays interface statistics in real time?
4. Describe switches that you can use with the ping and traceroute utilities.
5. What command will search a long file for instances of the word "fail"? How could you easily count the number of such instances?
6. What is the purpose of a snapshot command?
7. Describe software packaging and upgrade procedures.



This Chapter Discussed:

- Monitoring platform and interface operation;
- Using network utilities;
- Configuring system logging and parsing log files for error symptoms;
- Managing licenses;
- Maintaining JUNOS software; and
- Performing file system maintenance and password recovery.

Lab 3: Operational Monitoring

- Use J-Web and the CLI to monitor and maintain a J-series platform.



Lab 3: Operational Monitoring

The slide shows the objectives for this lab.



Operating Juniper Networks Routers in the Enterprise

Chapter 6: Routing Protocols and Policy

Chapter Objectives

- After successfully completing this chapter, you will be able to:
 - Describe routing tables and route preferences
 - Describe the role of JUNOS software routing policy
 - Describe J-Web support for routing protocols and policy
 - Configure and monitor static routes
 - Explain the role of IGPs
 - Configure and monitor basic RIP
 - Configure and monitor basic OSPF
 - Explain the purpose of exterior gateway protocols
 - Configure and monitor basic BGP



This Chapter Discusses:

- Routing tables and route preferences;
- JUNOS software routing policy and monitoring its operation;
- Static routing;
- An overview of interior gateway protocol (IGP) operation and purpose;
- RIP configuration and operation;
- OSPF configuration and operation; and
- BGP overview and basic configuration.

Agenda: Routing Protocols and Policy

- Routing Table and Route Preferences
- Routing Policy
- J-Web Support for Routing Protocols and Policy
- Configuring and Monitoring Static Routing
- Interior Gateway Protocols
- Configuring and Monitoring RIP
- Configuring and Monitoring OSPF
- Configuring and Monitoring Basic BGP

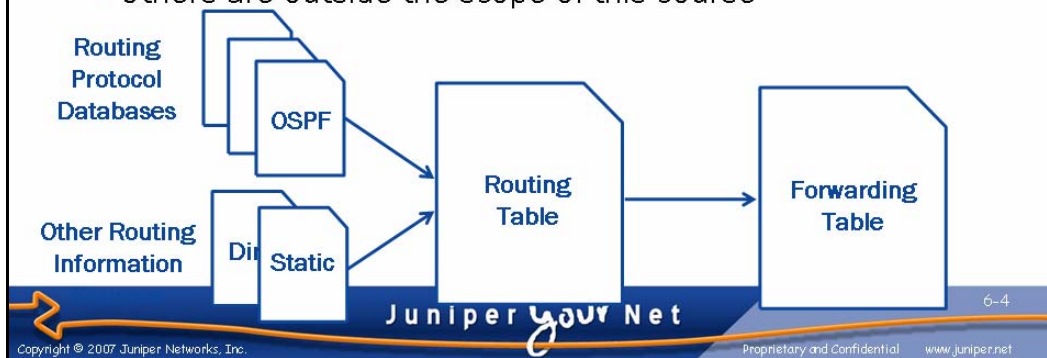


Routing Tables and Route Preferences

The slide lists the topics we cover in this chapter. We discuss the highlighted topic first.

Routing Tables

- Compiles information learned from routing protocols and other routing information sources
- Selects an active route to each destination
- Populates forwarding table
- Multiple routing tables
 - inet.0 for IPv4 unicast routing
 - Others are outside the scope of this course



Routing Information Sources

The JUNOS software routing table consolidates prefixes from multiple routing information sources including various routing protocols, static routes, and directly connected routes.

Active Route Selection

When multiple routes are received for a given prefix, a single route is selected as the active route. Multiple, equal-cost routes are supported with additional configuration.

Forwarding Table

The active route for each destination is used to populate the router's forwarding table. The forwarding table determines the outgoing interface and Layer 2 rewrite information for each packet the router forwards.

Multiple Routing Tables

Juniper Networks routers actually maintain multiple routing tables. The primary routing table, inet.0, is used to store IPv4 unicast routes. Additional tables are used for purposes such as multicast, IPv6, MPLS traffic engineering, and VPNs. This course concentrates solely on the inet.0 routing table.

Route Preference

- Ranks routes received from different sources
- Primary criterion for selecting the active route
- Ranges from 0 to 4,294,967,295, with lower value preferred

Route Preference Values

Routing Information Source	Default Preference
Direct	0
Local	0
Static	5
OSPF internal	10
RIP	100
Aggregate	130
OSPF AS external	150
BGP (both EBGp and IBGP)	170

Preferred Routing Information Sources

Route preference is used to differentiate routes received from different routing protocols or routing information sources. Route preference is equivalent to administrative distance on other vendors' equipment.

Primary Tiebreaker

JUNOS software uses route preference as the primary criterion for selecting the active route. Preference values cause routes from certain information sources to be ranked more preferable than the same route received from another information source. The table at the bottom of the slide shows the default preference values for a selected set of routing information sources.

Continued on next page.

Lower Is Better

Routing preference values can range from 0 to 4,294,967,295. Lower preference values are preferred over higher preference values. This command output demonstrates that a direct route with a preference of 0 is preferred over an OSPF internal route with a preference of 10:

```
user@host> show route 10.251.254.130/31 exact
```

```
inet.0: 18 destinations, 19 routes (17 active, 0 holddown, 1 hidden)  
+ = Active Route, - = Last Active, * = Both
```

```
10.251.254.130/31  *[Direct/0] 1d 07:53:39  
                   > via tl-4/0/0.0  
                   [OSPF/10] 1d 07:53:32, metric 65  
                   > via tl-4/0/0.0
```

Agenda: Routing Protocols and Policy

- Routing Table and Route Preferences
- Routing Policy
- J-Web Support for Routing Protocols and Policy
- Configuring and Monitoring Static Routing
- Interior Gateway Protocols
- Configuring and Monitoring RIP
- Configuring and Monitoring OSPF
- Configuring and Monitoring Basic BGP



Routing Policy

The slide highlights the topic we discuss next.

Routing Policy Overview

- Controls routing information transferred into and out of the routing table
 - Can ignore or change incoming routing information
 - Can suppress or change outgoing routing information
- Policies are made up of match/action pairs
 - Match conditions can be protocol specific
- Apply policy when:
 - You do not want to import all learned routes into the routing table
 - You do not want to advertise all learned routes to neighboring routers
 - You want one protocol to receive routes from another protocol
 - You want to modify information associated with a route



Concept of Routing Policy

The concept of routing policy has been around for many years and is not specific to Juniper Networks platforms. Policy is a very powerful tool that lets you manipulate routes that you receive, send, or both. In other words, you can manipulate the default route selection process of the router by changing route attributes or ignoring and suppressing routes. As we look at policy in more detail, note that policy evaluation is centered on the routing table. Subsequent slides address this fact.

Match/Action Pairs

JUNOS software policies are sets of match and action pairs. The match section is a listing of criteria; the action section defines what to do if the match criteria are satisfied. For those familiar with programming, this concept is similar to an if/then statement.

Continued on next page.

Applying Policy

Generically speaking, you use JUNOS software policies when you want to alter the default behavior of the router. More specifically, you might want to filter routing information from a neighbor, filter routes to a neighbor, or redistribute routes between routing protocols.

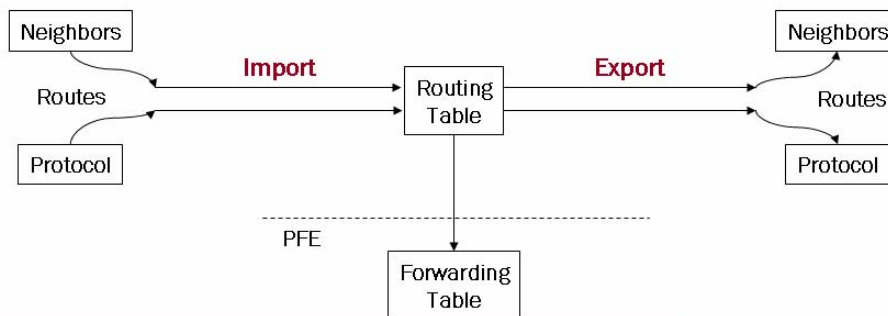
The filtering of routing information is one major use of the policy language. Based on criteria such as protocols or individual routes, you have the ability to allow or deny information to neighboring routers.

If a situation exists in your networking environment where information from a particular protocol (such as static routes) must be sent to another protocol (such as BGP), you need a policy. Due to the match/action pairing within a policy, you can select the criteria of *all static routes* and the action to perform of *send out via BGP* with relative ease.

Lastly, you can alter and modify attribute information within the routes by using a policy. You can change things such as metric values and JUNOS software route preference.

Import and Export Policies

- Perform policy filtering with respect to the JUNOS software routing table
 - JUNOS software applies import policy prior to inclusion in the routing table
 - JUNOS software applies export policy *only* to active routes in the routing table

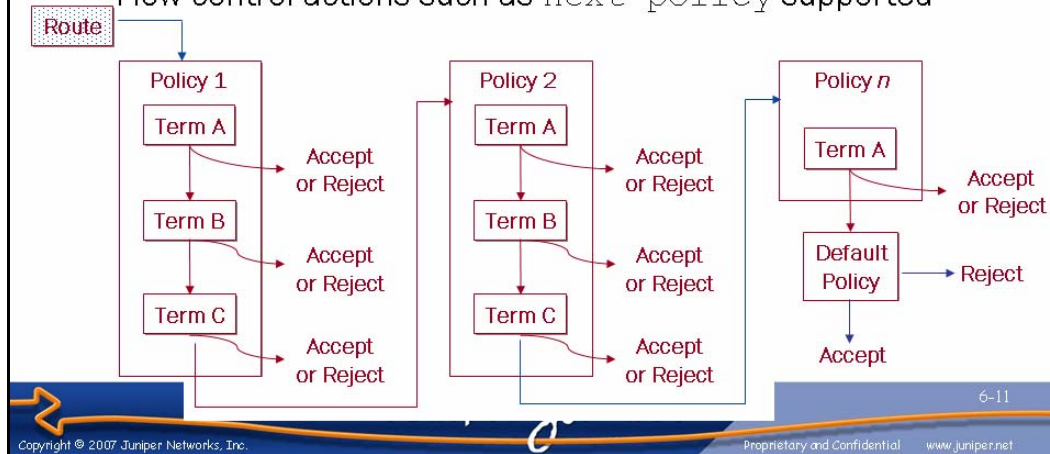


Policy Filtering

All policy processing on Juniper Networks J-series platforms occurs with respect to the routing table. JUNOS software applies policies as the routing table adds and removes routing information. The keywords *import* and *export* imply the direction of data flow with respect to the routing table.

Routing Policy Flow

- Policies can be chained together
 - Evaluation proceeds left to right until a *terminating* action of accept or reject is reached
- Individual policies can contain a collection of terms
 - Flow-control actions such as `next-policy` supported



Policy Chaining

Policies can be cascaded to form a chain of policy processing. Creating this chain of policies is often done to solve a complex set of route manipulation tasks in a modular manner.

JUNOS software evaluates policies from left to right based on the order in which they are applied to a routing protocol. JUNOS software checks each policy's match criteria and performs the associated action when a match occurs. If the first policy does not match or if the match is associated with a nonterminating action, the route is evaluated against the next policy in the chain. This pattern repeats itself for all policies in the chain. JUNOS software ultimately applies the default policy for a given protocol when no terminating actions occur while evaluating the user-defined policy chain.

Policy processing stops once a route meets a terminating action, unless you are grouping policies with Boolean operators. Grouping policies for logical operations, such as AND or OR, is a subject that is beyond the scope of this class.

Continued on next page.

Individual Policies

Individual policies can be comprised of multiple entries called terms. Terms are individual match/action pairs and can be named numerically or symbolically.

JUNOS software lists terms sequentially from top to bottom and evaluates them in that manner. Each term is checked for its match criteria. When a match occurs, JUNOS software performs the associated action. If no match exists in the first term, JUNOS software checks the second term. If no match exists in the second term, JUNOS software checks the third term. This pattern repeats itself for all terms. If no match exists in the last term, JUNOS software checks the next applied policy.

When a match is found within a term, JUNOS software takes the corresponding action. When that action is taken, the processing of the terms and the applied policies stops.

Default Policies

- Protocols are associated with a default policy
- OSPF:
 - Import: Accept all LSAs flooded by that protocol
 - Export: Reject everything
 - LSA flooding announces OSPF-learned and local routes
- RIP:
 - Import all learned RIP routes, export nothing
 - RIP requires export policy to announce RIP (or other) routes
- BGP:
 - Import all routes learned from BGP neighbors
 - Export all active routes learned from BGP neighbors to all BGP neighbors
 - EBGp-learned routes are exported to all BGP peers
 - IBGP-learned routes are exported to all EBGp peers (assumes logical IBGP full mesh)



Default Policy

The default policy always applied to a string of policies sounds very mysterious, but in reality it is not. In fact, every routing protocol that runs on a Juniper Networks J-series platform always applies the default policy for that protocol. Simply put, the default policy is the default operation of the protocol.

You can override the default action intrinsic to a particular protocol by including a `default-action [accept | reject]` within a policy statement. The `default-action` statement is a nonterminating action modifier, which means that subsequent policy statements can continue to evaluate matching routes.

OSPF

For IGP's such as OSPF and IS-IS, the default import policy is to accept all routes learned from that protocol. Technically speaking, link-state protocols do not receive routes. Instead, link-state information is flooded to all routers to create a link-state database. Each router then computes optimal paths from this database using a shortest-path-first (SPF) algorithm. The default export policy rejects *all* routes; this is because these protocols advertise routes learned through that protocol, and local routes, by flooding link-state information. Using an export policy to limit link-state advertisement (LSA) flooding would break the operation of a link-state protocol.

Continued on next page.

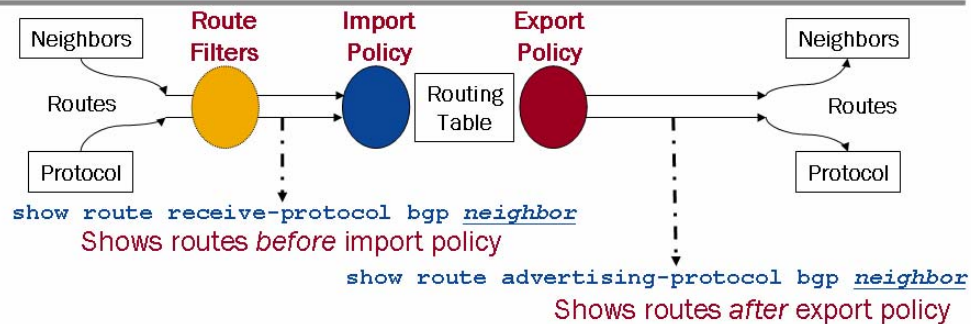
RIP

The default RIP import policy is to accept all routes learned through RIP. The default export policy advertises no routes, not even those learned through RIP.

BGP

The default BGP import policy has all received BGP routes imported into the routing table. For export, all active BGP routes are sent to all peers, with the exception of not sending routes learned through internal BGP (IBGP) to other IBGP speakers. This behavior is in accordance with BGP protocol requirements.

Monitoring Policy Operation



- The `show route receive-protocol` and `show route advertising-protocol` CLI commands:
 - Display routing updates received *before* import and *after* export policy processing, respectively
 - Filtered routes are the exception for import policy
- Pop quiz: How can you monitor the effects of an import policy?

Monitoring Effects of Policy

The commands on the slide show routing updates received before import policy processing and the routing updates sent after export policy processing.

Use the `show route receive-protocol protocol neighbor` command to show the specified protocol-type route advertisements that a particular neighbor is advertising to your router before import policy is applied. Use the `show route advertising-protocol protocol neighbor` command to show the protocol-type route advertisements that you are advertising to a particular neighbor after export policy is applied.

The use of route filters marks an exception to the behavior documented previously. JUNOS software evaluates route filters before the output of a `show route receive-protocol` command is generated. Thus, you must specify the hidden switch to the `show route receive-protocol` command to display received routes filtered by your import policy.

Answer

After import policy processing, use the `show route protocol protocol` command to monitor the effects of your import policy. This command shows all routes from the protocol type specified that are installed in the routing table.

Agenda: Routing Protocols and Policy

- Routing Table and Route Preferences
- Routing Policy
- J-Web Support for Routing Protocols and Policy
- Configuring and Monitoring Static Routing
- Interior Gateway Protocols
- Configuring and Monitoring RIP
- Configuring and Monitoring OSPF
- Configuring and Monitoring Basic BGP

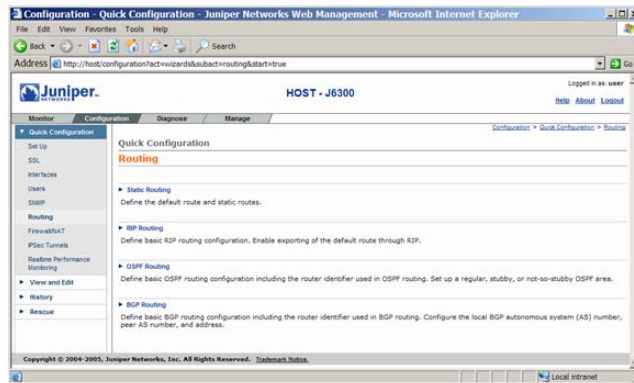


J-Web Support for Routing Protocols and Policy

The slide highlights the topic we discuss next.

J-Web and Routing Protocols

- J-Web routing protocol wizards found at Configuration > Quick Configuration > Routing and Protocols
 - Quickly establish basic connectivity for:
 - Static, RIP, OSPF, and BGP routing



Routing Protocol Wizards

J-Web simplifies configuration of static routing as well as basic configuration of the RIP, OSPF, and BGP routing protocols. You can access these routing protocol wizards at the J-Web Configuration > Quick Configuration > Routing and Protocols page.

J-Web and Routing Protocols

- Use J-Web configuration editor (or the CLI) to:
 - Tweak OSPF default route origination, summarization, authentication, etc.
 - Create and apply routing policy



Advanced Routing Configuration

While the J-Web wizards are great for basic routing protocol configurations, they do not support advanced configurations. Use the J-Web Configuration > View and Edit > Edit Configuration functionality or the CLI to configure advanced features. Do not forget to commit your changes.

Monitoring Routing with J-Web

- Use J-Web to monitor routing at the Monitor > Routing page

Displays the routing table

Displays protocol-specific information

J-Web Route Monitoring

You can use the J-Web Monitor > Routing page to display the routing table or protocol-specific routing information.

Sample J-Web Route Table Display

Table name and summary

Route table contents

Filter display using these fields

Destination	Protocol/Preference	Next-Hop	Age
10.0.0.1/32	*OSPF/10	to 10.251.254.137 via fe-1/0/1.0, selected	1d 8:56:11
10.251.254.130/31	OSPF/10	via ti-4/0/0.0, selected	1d 8:56:56
10.251.254.132/30	*OSPF/10	to 10.251.254.146 via fe-1/0/0.0, selected	1d 8:56:11
10.251.254.140/31	*OSPF/10	to 10.251.254.137 via fe-1/0/1.0, selected	1d 8:56:11
10.251.254.252/32	*OSPF/10	to 10.251.254.146 via fe-1/0/0.0, selected	1d 8:56:16
10.251.254.253/32	*OSPF/10	to 10.251.254.137 via fe-1/0/1.0, selected	1d 8:56:11
224.0.0.5/32	*OSPF/10		1d 8:57:51

Narrow Search

Destination Address: Protocol:

Next Hop Address: Receive Protocol:

Best Route: ☐ Inactive Routes: ☐

Exact Route: ☐ Hidden Routes: ☐

Number of Routes to Display:

OK

Viewing the Routing Table with J-Web

The J-Web Monitor > Routing > Route Information page allows you to view entries in the routing table. The display shows the routing table name and summary information that includes the total number of routes in the table. The next section of the page includes a terse display of each routing table entry. You can click the plus sign (+) next to an entry to obtain more detailed information about the route.

The Narrow Search section of the page allows you to display only a subset of entries in the routing table. The slide demonstrates using this feature to display only the OSPF-learned routes. You can also filter the display based on other criteria such as the route destination or next hop.

From the CLI you can obtain equivalent information with the **show route** command and its various switches. For example, **show route protocol ospf** provides the same information as shown in the J-Web screen capture.

Note that the 10.251.254.130/32 route in this display is inactive because it is an OSPF route; there happens to be a better route that is not currently visible because the display is only showing OSPF routes.

Agenda: Routing Protocols and Policy

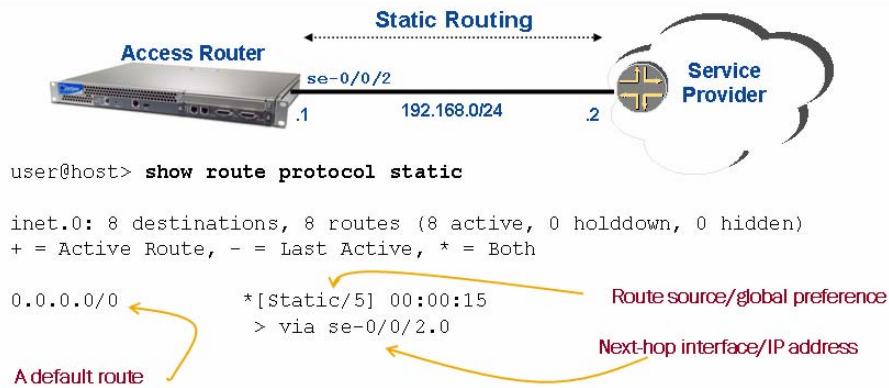
- Routing Table and Route Preferences
- Routing Policy
- J-Web Support for Routing Protocols and Policy
- ➔ **Configuring and Monitoring Static Routing**
- Interior Gateway Protocols
- Configuring and Monitoring RIP
- Configuring and Monitoring OSPF
- Configuring and Monitoring Basic BGP



Configuring and Monitoring Static Routing

The slide highlights the topic we discuss next.

Static Routing



- Static routing is often used when single-homed to a service provider
 - Static default route directs external traffic to the service provider

Static Routes

You can use static routes in a networking environment for multiple purposes, including a default route for the autonomous system (AS) as well as routes to customer networks. Unlike dynamic routing protocols, you manually configure the routing information provided by static routes on each router in the network.

By default, the next-hop IP address of static routes configured in the JUNOS software must be reachable via a direct route. Unlike other vendors, recursive lookups of next hops are not performed by default.

Static routes remain in the routing table until you remove them or they become *nonactive*. One possible way for a static route to be nonactive is for the IP address of the next hop to be unreachable across a directly connected interface.

Continued on next page.

Static Routes (contd.)

All configuration for static routes occurs at the `[edit routing-options static]` level of the hierarchy. Attributes that you can associate with a static route include the following:

- `as-path`: Used if this route is intended to be redistributed into BGP and you want to add values manually to the AS-path attribute.
- `community`: Used if this route is intended for BGP and you want to add community values to the route for use in your AS.
- `metric`: If multiple routes share the same preference value, the route with the best metric becomes active in the routing table. Use this value to prefer one route over another in this case.
- `preference`: The default preference value of static routes is 5. This preference makes them more likely to be active than OSPF, IS-IS, or BGP for matching prefixes. Use this option to increase the value of the static routes to prefer other sources of routing information.

Static Routing Case Study

■ Use static routing to provide connectivity among all WAN, LAN, and loopback addresses

Juniper your Net

6-24

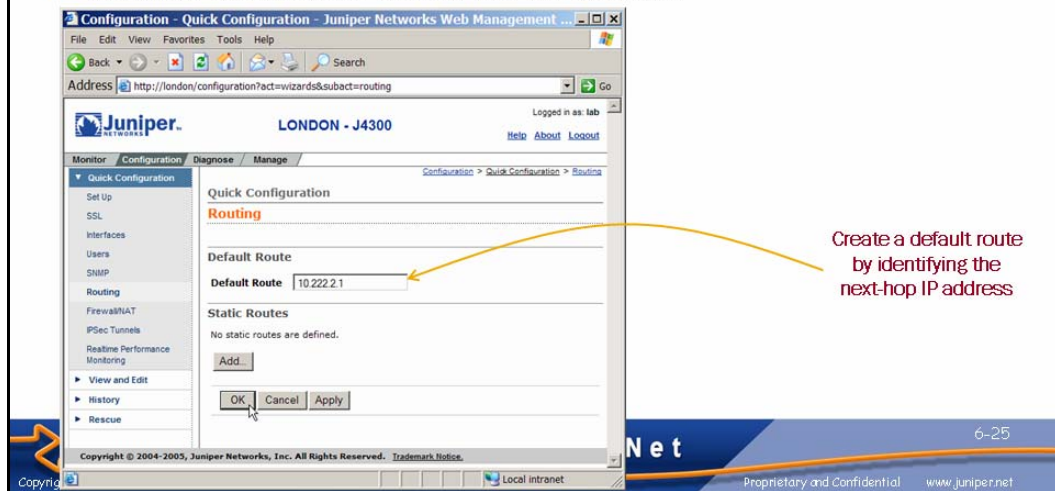
Copyright © 2007 Juniper Networks, Inc. Proprietary and Confidential www.juniper.net

Static Routing Example

We use the topology shown on the slide to demonstrate static routing. We will configure static routes to allow *Tokyo* to reach *London*'s 10.222.3.0/24 network and 192.168.36.1/32 loopback address. Likewise, we will configure *London* with a static default route that allows it to reach all destinations through its interface to *Tokyo*.

Default Route Configuration

- Access the J-Web static routing wizard at the Configuration > Quick Configuration > Routing and Protocols page
- Create a default route on *London*



Default Route Definition

A static default route is configured at the *London* router using the J-Web static routing wizard. This route points to *Tokyo*'s *se-1/0/0* interface IP address. The resulting CLI configuration is as follows:

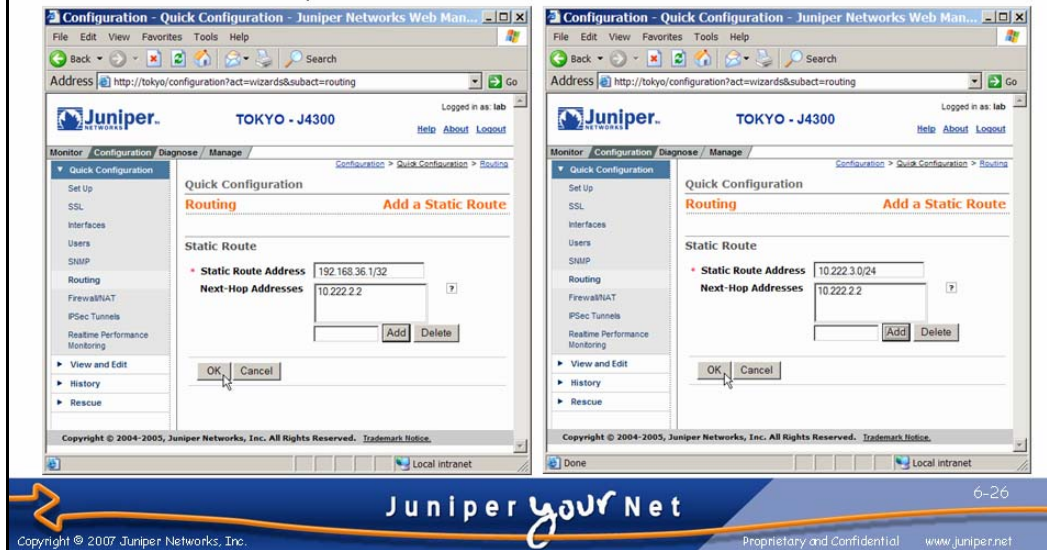
```
[edit routing-options]
lab@London# show
static {
    route 0.0.0.0/0 next-hop 10.222.2.1;
}
```

The equivalent CLI command is revealed when the configuration stanza is piped to **display set**:

```
[edit routing-options]
lab@London# show | display set
set routing-options static route 0.0.0.0/0 next-hop 10.222.2.1
```

Static Route Configuration

- Static route definitions at *Tokyo*
 - Provides reachability to *London*'s loopback address and 10.222.3.0/24 network



Static Route Definition

Two static routes are configured at *Tokyo* using the J-Web static routing wizard. The resulting CLI configuration is as follows:

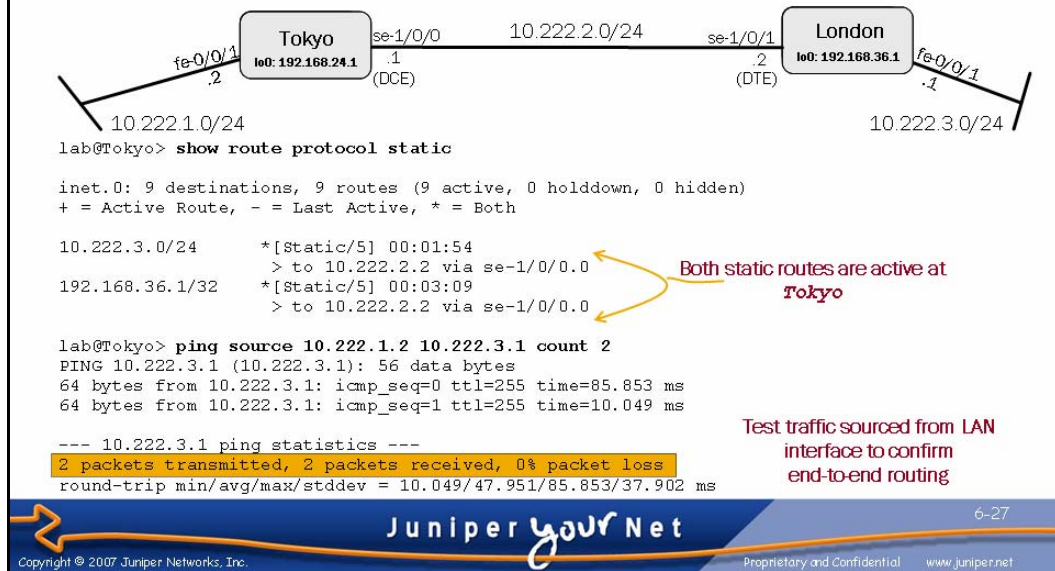
```
[edit routing-options]
lab@Tokyo# show
static {
    route 192.168.36.1/32 next-hop 10.222.2.2;
    route 10.222.3.0/24 next-hop 10.222.2.2;
}
```

The equivalent CLI commands are revealed when the configuration stanza is piped to **display set**:

```
[edit routing-options]
lab@Tokyo# show | display set
set routing-options static route 192.168.36.1/32 next-hop 10.222.2.2
set routing-options static route 10.222.3.0/24 next-hop 10.222.2.2
```

Confirming Static Routing

- Use J-Web or the CLI to display the routing table and to confirm reachability



Monitoring Static Routing

We use the **show route protocol static** CLI command at *Tokyo* to confirm that both static routes are installed in the routing table. We then use the **ping** command to confirm reachability between *Tokyo*'s fe-0/0/1 interface and *London*'s fe-0/0/1 interface. We accomplish this by sourcing the ping from *Tokyo*'s fe-0/0/1 interface IP address and choosing *London*'s fe-0/0/1 IP address as the destination.

Agenda: Routing Protocols and Policy

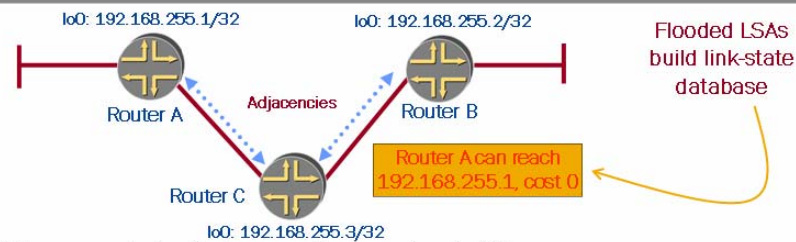
- Routing Table and Route Preferences
- Routing Policy
- J-Web Support for Routing Protocols and Policy
- Configuring and Monitoring Static Routing
- Interior Gateway Protocols
 - Configuring and Monitoring RIP
 - Configuring and Monitoring OSPF
 - Configuring and Monitoring Basic BGP



Interior Gateway Protocols

The slide highlights the topic we discuss next.

Interior Gateway Protocol Overview



- IGP provide internal reachability
 - Promote connectivity but lack administrative controls needed to enforce routing policy
- Normally, link-state routing (OSPF) is deployed
 - Optimal convergence and bandwidth usage based on reliable flooding of link-state updates
 - Builds a replicated network topology database at all stations within an OSPF area or IS-IS level and uses SPF to find optimal paths
 - RIP and static routing are also common

IGPs Provide Internal Reachability

The purpose of an IGP is to provide optimal reachability to destinations that lie within a particular routing domain. Note that an AS can be comprised of one or more IGP routing domains, depending upon the technical and political issues that held sway when the network was designed. In most service provider networks, the IGP does not carry customer or external routes; one of the main jobs of the IGP is to promote IBGP peering between the loopback addresses of the BGP-speaking routers in the network. Loopback peering is preferred for stability and reliability purposes, and without the services of an IGP, routers would be unable to reach each other's loopback address, resulting in IBGP session establishment failures.

It can be said that an IGP lives to provide connectivity at any cost. While this is a likable enough trait, the lack of administrative controls and desire to connect everything makes an IGP unsuitable for the purpose of enforcing an AS's routing policy. This is where BGP (covered later in the section) comes into play.

Continued on next page.

Link State or Bust

Virtually all service provider networks deploy IGPs that are based on the concept of link-state routing, as opposed to their older, and generally less optimal, distance-vector cousins.

A link-state routing protocol is based on the tenets of the reliable flooding of link-state packets that describe the originating router's interfaces and reachability costs. Because these link-state updates are originated by all routers in a given area or IS-IS level, and because they are reliably flooded, the net result is that all routers build a replicated database describing the network's topology. An SPF algorithm is then run against the database to select optimal (shortest) paths to each internal destination.

Agenda: Routing Protocols and Policy

- Routing Table and Route Preferences
- Routing Policy
- J-Web Support for Routing Protocols and Policy
- Configuring and Monitoring Static Routing
- Interior Gateway Protocols
- Configuring and Monitoring RIP
- Configuring and Monitoring OSPF
- Configuring and Monitoring Basic BGP



Configuring and Monitoring RIP

The slide highlights the topic we discuss next.

What Is RIP?

- RIP is an IGP that is used within an AS
- Two versions:
 - RIPv1 (RFC 1058)
 - RIPv2 (RFC 2453)
- Primary characteristics:
 - Distance-vector routing protocol; prone to loops and slow convergence
 - Split horizon and poison reverse for loop prevention
 - Hop count is used as the metric for path selection, based on Bellman-Ford distance-vector routing algorithm
 - Routing updates sent every 30 seconds



RIP Is an Interior Gateway Protocol

RIP is an IGP used *within* an AS. RIP advertises routes between devices within the AS.

Two Versions

Two versions of RIP exist: RIPv1 and RIPv2. RIPv2 did not change the protocol; it expanded the capabilities of it. RFC 1058 defines RIPv1; RFC 2453 defines RIPv2. RIPv1 and RIPv2 can interoperate if RIPv1 ignores all fields that must be zero. RIPv2 allows more information to be included in RIP packets and provides a simple authentication mechanism. It also supports variable-length subnet mask (VLSM).

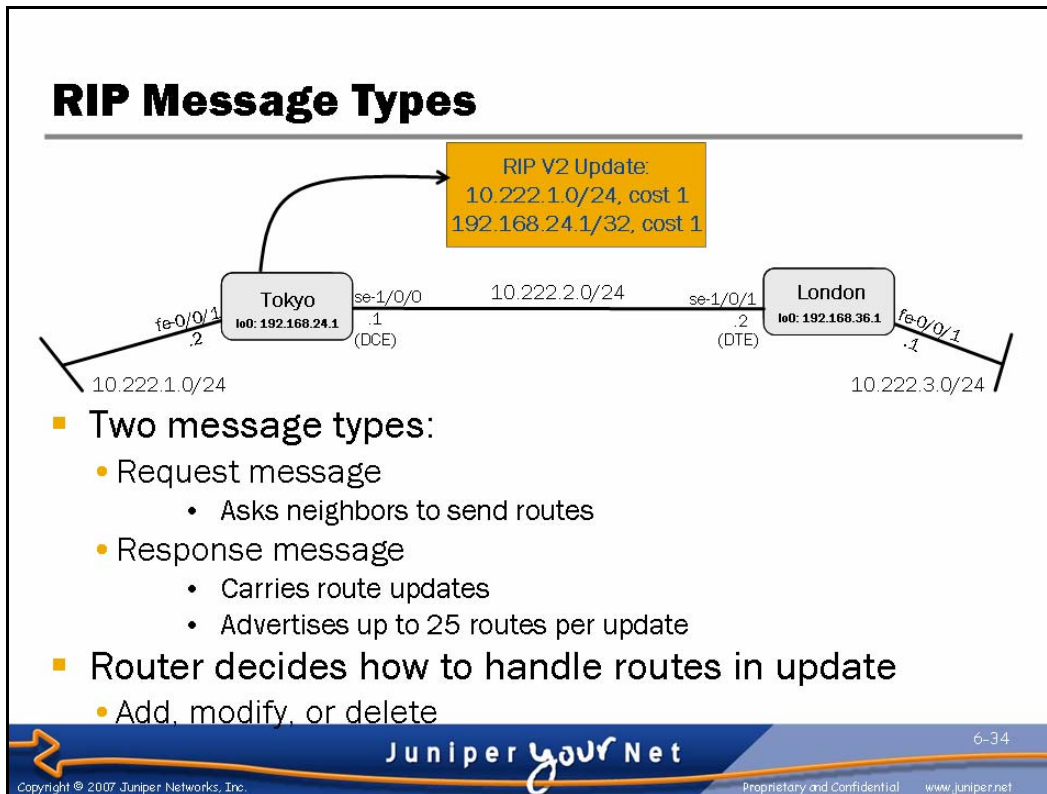
RIP is based on the ROUTED program, originally distributed with version 4.3 of the Berkeley Software Division UNIX software. In most UNIX systems, the ROUTED routing process dynamically builds the routing table based on information it receives through RIP updates. When routing starts, it issues a request for routing updates and then listens for responses to its request. When a system configured to supply RIP information hears the request, it responds with an update packet based on the information in its routing table. The update packet contains the destination addresses from the routing table and the routing metric associated with each destination. Update packets are not just issued in response to requests, they are also issued periodically to keep routing information accurate.

Continued on next page.

Characteristics

The following list details the primary characteristics of RIP:

- *Distance-vector*: When determining the best path to a destination, RIP uses a combination of hop count (that is, distance) and the next hop (that is, vector).
- *Hop count*: The longest network path in an RIP network is 15 hops between the source and the destination. The assumption here is that the metric count for each network or hop has a cost of one. The 15-hop limitation exists to prevent the creation of an infinitely long network path. With an upper limit of 15 hops, the protocol treats a metric of 16, referred to as *infinity*, to mean that the destination network is unreachable, referred to as *network unreachable*.
- *Periodic updates*: Upon receiving an update from another router, the requesting router validates the response and might or might not update its routing table. If updating is required, the update can take the form of adding a route to the table, modifying an existing entry, or deleting an existing entry. Upon receipt of all replies from connected routers, the requesting router builds and updates its routing table. Every 30 seconds, RIP sends all or part of the router's routing table to each of its neighbor's directly connected routers. The routing table is either broadcast to its neighbors on an Ethernet segment or sent to the other end of a point-to-point link. These periodic updates allow a router running RIP to respond to network changes. RIP also supports triggered updates. A triggered update occurs when a metric changes on a route and can include only the changed entry or entries. Each entry in the routing table consists of the following:
 - Network reachability information, the network ID, and the metric;
 - Next-hop information;
 - The interface through which a packet must pass; and
 - A timer indicating the age of a routing entry.



Update Process

A request message asks neighboring routers to send an update, and a response message carries the update from the neighboring routers. When a router receives an update from a neighbor, RIP adds the cost of the network over which the update is received to the advertised metric. The new value is used when comparing routes. RIP stores unknown routes immediately. If a router must advertise more than 25 routes, it must send out an additional response message.

Continued on next page.

Route Updates

RIP evaluates known routes by comparing the metric, or cost, of the route presently in the table to the metric of the received route with the following decisions:

- If the cost is lower, RIP adds the new route to the table.
- Where the router advertising the network is the same as that which originally provided it, RIP adopts the route, even where the metric is larger.
- If the advertised hop count is higher than the recorded hop count and the recorded next-hop router originated the update, RIP marks the route as unreachable for a specific hold-down period. At the end of the hold-down period, if the same neighbor is still advertising the higher hop count, RIP accepts the new metric.

The router can receive both RIPv1 and RIPv2 update messages, with 25 route entries per message. RIP uses timers to enable the router to make the decisions described previously.

RIPv2 Features

The 192.168.1.0 prefix is subnetted with a variable-length netmask

Update: 192.168.1.192/26, Cost 1

RIP V2 updates include the netmask in updates to support VLSM

- Backward compatible with RIPv1
- Update includes prefix length to support VLSM
- Authentication on a per-message basis
 - Simple password or MD5 authentication
- Updates sent to multicast address 224.0.0.9
 - Broadcast-based updates can be configured

Copyright © 2007 Juniper Networks, Inc. Proprietary and Confidential www.juniper.net

Backward Compatibility

RIPv2 is totally *backward compatible* with RIPv1. If a RIPv2 router receives a RIPv1 request message, it should respond with a RIPv1 response message. If you configure the router to send only RIPv2 messages, it should not respond to a RIPv1 request message.

Prefix Length

RIPv2 can perform *classless routing*, where the prefix length is included in the RIP updates. Another benefit of having a destination prefix length associated with an update is that you can use *variable-length destination prefixes*, thus eliminating the requirement that all destination prefixes in the Internet have the same length.

Authentication per Message

Authentication is possible with RIPv2. The authentication scheme for RIPv2 uses the space of an entire RIP entry. If the address family identifier of the first—and only the first—entry in the message is 0xFFFF, the remainder of the entry contains the authentication. Thus, at most, 24 RIP entries in the remainder of the message can exist. If authentication is not in use, no entries in the message should have an address family identifier of 0xFFFF. Currently, the only *authentication types* are simple password and Message Digest 5 (MD5). Simple passwords use Type 2 and MD5 uses Type 3.

Continued on next page.

Multicast Updates

Multicasting was added to reduce unnecessary processing of RIP updates by hosts who are not involved in RIPv2 processing. The IP multicast address is 224.0.0.9. On nonbroadcast multiaccess networks, like Frame Relay or ATM, you can use unicast addressing.

RIP Limitations

■ Limitations:

- Maximum network diameter = 15 hops
- Regular updates include entire routing table approximately every 30 seconds
- Poison reverse increases size of routing updates
- Count to infinity slows route-loop prevention
- Metrics reflect hop count only
- Broadcasts between neighbors (RIPv1 only)
- Classful routing means no prefix length carried in route updates (RIPv1 only)
- No authentication mechanism (RIPv1 only)
- Poor convergence



RIP Limitations

RIP's limitations include the following:

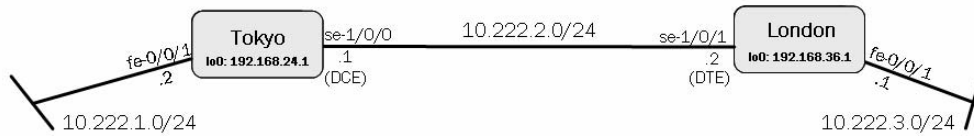
- The designers believe that the basic protocol design is inappropriate for larger networks. Assuming a cost of 1, the protocol is limited to networks whose longest path involves 15 hops. If we choose to use larger costs, the upper bound of 15 can become a problem easily.
- Routing updates occur every 30 seconds, and the entire routing table is sent in an update. In addition, a triggered update, resulting from a network change, occurs immediately and involves sending the entire routing table.
- Poison reverse aids in network convergence, but it also increases the size of update messages, which include valid and poisoned routes.
- The protocol depends upon counting to infinity to resolve certain unusual situations. Resolving a loop with counting to infinity involves time because a route's metric is increased by two in each update interval, and the loop is only broken when the count reaches 16.
- This protocol uses fixed metrics to compare alternative routes. This method is not appropriate, however, for situations where routes must be chosen based on real-time parameters, such as measured delay, reliability, or load.
- Broadcasting between neighbors forces processing of packets by each host, whether involved in the routing process or not.

Continued on next page.

RIP Limitations (contd.)

- RIPv1 cannot distinguish between subnets. RIPv1 cannot advertise destination prefix lengths; thus, all networks involved in an RIPv1 network must use the same mask.
- RIPv1 provides no authentication mechanism, so a RIP router accepts all RIP-compliant updates.
- Convergence on the network can be slow, leading to loops and suboptimal paths.

RIP Case Study



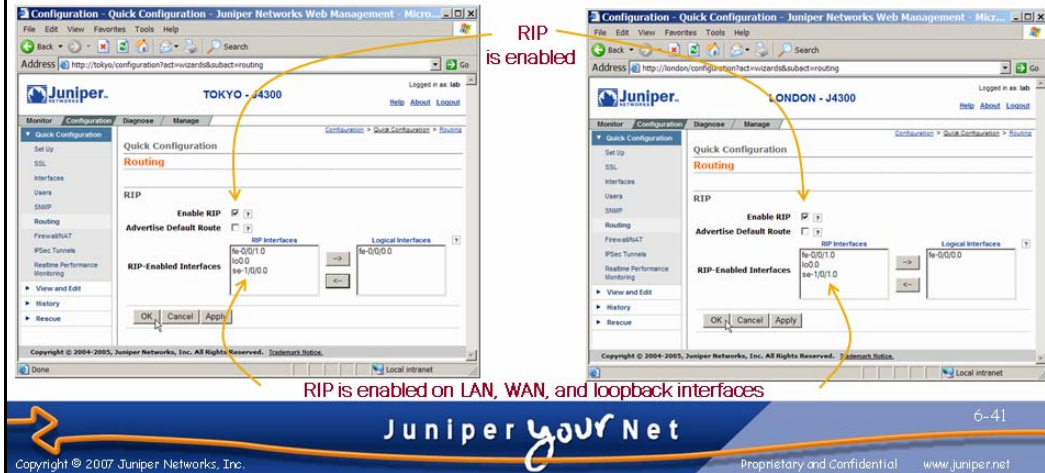
- Use RIPv2 to provide connectivity among all WAN, LAN, and loopback addresses

RIPv2 Example

We use the topology shown on the slide to demonstrate RIP routing. We will enable RIP on the WAN, LAN, and loopback interfaces of both *Tokyo* and *London*. This setup will provide connectivity between all networks in the topology.

Configuring RIP: J-Web

- Use the J-Web Configuration > Quick Configuration > Routing and Protocols > RIP Routing wizard
 - Automatically creates export policy to advertise RIP-enabled interface and learned RIP routes



J-Web RIP Routing Wizard

The J-Web Configuration > Quick Configuration > Routing and Protocols > RIP Routing wizard greatly simplifies configuration of the RIP protocol. You simply check the Enable RIP box and select the logical interfaces that should participate in the protocol. Like all wizards, clicking OK or Apply commits your changes.

Remember that the default RIP export policy does not advertise any routes, including those learned from RIP. The J-Web wizard, however, automatically creates a policy to advertise direct routes on RIP-enabled interfaces and RIP-learned routes.

The Resulting RIP Configuration

```
lab@London# show protocols rip
group jweb-rip {
  export [ jweb-policy-rip jweb-policy-direct ];
  neighbor fe-0/0/1.0;
  neighbor lo0.0;
  neighbor se-1/0/1.0;
}

[edit]
lab@London# show policy-options
policy-statement jweb-policy-rip {
  from protocol rip;
  then accept;
}
policy-statement jweb-policy-direct {
  from {
    protocol direct;
    interface [ fe-0/0/1.0 lo0.0 se-1/0/1.0 ];
  }
  then accept;
}
```

Two export policies are in effect

Export policies override default behavior by advertising RIP interfaces and learned RIP routes



The Results

The configuration that results from the J-Web RIP wizard is piped through **display set** to reveal the corresponding CLI syntax:

```
[edit]
lab@London# show protocols rip | display set
set protocols rip group jweb-rip export jweb-policy-rip
set protocols rip group jweb-rip export jweb-policy-direct
set protocols rip group jweb-rip neighbor fe-0/0/1.0
set protocols rip group jweb-rip neighbor lo0.0
set protocols rip group jweb-rip neighbor se-1/0/1.0

[edit]
lab@London# show policy-options | display set
set policy-options policy-statement jweb-policy-rip from protocol rip
set policy-options policy-statement jweb-policy-rip then accept
set policy-options policy-statement jweb-policy-direct from protocol direct
set policy-options policy-statement jweb-policy-direct from interface fe-0/0/1.0
set policy-options policy-statement jweb-policy-direct from interface lo0.0
set policy-options policy-statement jweb-policy-direct from interface se-1/0/1.0
set policy-options policy-statement jweb-policy-direct then accept
```

Monitoring RIP: J-Web

- Use the J-Web Monitor > Routing > RIP Information page to monitor general RIP operation

The screenshot shows the Juniper J-Web Monitor interface for a router named 'LONDON - J4300'. The left sidebar contains a navigation menu with categories like System, Chassis, Interfaces, Routing, Service Sets, Firewall, IPSec, NAT, and RPM. The 'Routing' category is expanded, showing sub-items like Route Information, BGP Information, OSPF Information, and RIP Information. The 'RIP Information' page is active, displaying the following data:

RIP Statistics

RIP Protocol Name	RIPv2
RIP Port	520
RIP Update Interval	30s
Hold Down	180s
Timeout	120s

Routes Learned

Routes Learned	Routes Held Down	Requests Dropped	Responses Dropped
2	0	0	0

RIP Neighbors

Neighbor	State	Source Address	Destination Address	Send Mode	Receive Mode	In Metric
fe-0/0/1.0	Up	10.222.3.1	224.0.0.9	mcast	both	1
lo0.0	Up	192.168.36.1	224.0.0.9	mcast	both	1
se-1/0/1.0	Up	10.222.2.2	224.0.0.9	mcast	both	1

Two red arrows point to the 'Routes Learned' table, indicating that two routes were learned via RIP. Another red arrow points to the 'RIP Neighbors' table, indicating that it shows RIP interface parameters.

Monitoring RIP with J-Web

The J-Web Monitor > Routing > RIP Information page allows you to monitor general RIP operation. It details overall RIP statistics, routes learned from RIP, and RIP interface configuration. Clicking an interface reveals RIP packet counters for that interface.

Monitoring RIP Using the CLI (1 of 3)

- Show the state of your RIP interfaces using the **show rip neighbor** command

```
lab@London> show rip neighbor
```

Neighbor	State	Source Address	Destination Address	Send Mode	Receive Mode	In Met
se-1/0/1.0	Up	10.222.2.2	224.0.0.9	mcast	both	1
lo0.0	Up	192.168.36.1	224.0.0.9	mcast	both	1
fe-0/0/1.0	Up	10.222.3.1	224.0.0.9	mcast	both	1

- Show routes learned via RIP using the **show route protocol rip** command

```
lab@London> show route protocol rip
```

```
inet.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
10.222.1.0/24      *[RIP/100] 00:21:16, metric 2, tag 0
                  > to 10.222.2.1 via se-1/0/1.0
192.168.24.1/32   *[RIP/100] 00:21:16, metric 2, tag 0
                  > to 10.222.2.1 via se-1/0/1.0
224.0.0.9/32      *[RIP/100] 00:21:22, metric 1
                  MultiRecv
```



State of RIP Interfaces

The **show rip neighbor** command lists interfaces currently running RIP. The output fields of this command are the following:

- Neighbor:** Displays the name of RIP neighbor.
- State:** Displays the state of the connection. The interface can be either up or down.
- Source Address:** Displays the source address.
- Destination Address:** Displays the destination of RIP updates, which can be either broadcast or multicast.
- Send Mode:** Displays the send options, which can be broadcast, multicast, none, or version 1.
- Receive Mode:** Displays the type of packets to accept, which can be both, none, version 1, or version 2.
- In Met:** Displays the metric added to incoming routes when advertising routes into RIP that were learned from other protocols.

RIP Routes

To view the routes in the unicast routing table, issue the **show route protocol rip** command. This command filters your routing table and shows only entries learned using RIP.

Monitoring RIP Using the CLI (2 of 3)

- Display RIP routes advertised out an interface using the `show route advertising-protocol rip neighbor` command
 - neighbor is the IP address of *local* RIP interface

```
lab@London> show route advertising-protocol rip 10.222.2.2

inet.0: 10 destinations, 10 routes (10 active, 0 holddown, 0
hidden)
+ = Active Route, - = Last Active, * = Both

10.222.3.0/24      *[Direct/0] 01:19:23
                  > via fe-0/0/1.0
192.168.36.1/32   *[Direct/0] 01:19:23
                  > via lo0.0
```

Advertisement of the LAN and loopback addresses owned by
London are confirmed on London's se-1/0/1 interface



Advertised RIP Routes

Use the `show route advertising-protocol rip neighbor` command to view the routes that are advertised out a RIP interface as a result of your RIP export policy. The neighbor argument in this command takes the form of the IP address assigned to the local router's RIP interface.

Note that to help guard against routing loops, the RIP protocol requires that a router continue to advertise a newly unreachable prefix with an infinite metric for a period of time after the route's status changes.

This poison reverse behavior can make it seem as though export policy changes are not taking effect because you might see the continued advertisement of prefixes that the current export policy should be rejecting when using the `show route advertising-protocol rip neighbor` command. When you adjust RIP export policy to reject routes previously being accepted, you should expect to see ongoing advertisement of the rejected prefixes for three RIP update cycles (approximately 90 seconds).

Monitoring RIP Using the CLI (3 of 3)

- Display RIP routes received on a particular interface using the `show route receive-protocol rip neighbor` command
 - neighbor is the IP address of *remote* RIP neighbor

```
lab@London> show route receive-protocol rip 10.222.2.1

inet.0: 10 destinations, 10 routes (10 active, 0 holddown, 0
hidden)
+ = Active Route, - = Last Active, * = Both

10.222.1.0/24      *[RIP/100] 00:27:04, metric 2, tag 0
                  > to 10.222.2.1 via se-1/0/1.0
192.168.24.1/32   *[RIP/100] 00:27:04, metric 2, tag 0
                  > to 10.222.2.1 via se-1/0/1.0
```

RIP advertisements for the LAN and loopback addresses owned by *Tokyo*
are confirmed on *London's* *se-1/0/1* interface



Received RIP Routes

Issue a `show route receive-protocol rip neighbor` command to view the routes being received on a RIP interface from the neighbor address specified. Note that the neighbor argument, in this case, is the IP address of the remote RIP neighbor.

Also note that the routes are displayed *before* your RIP import policy has a chance to manipulate their attributes, but *after* rejected routes are discarded due to unfavorable metrics or filtering. To confirm the operation of your RIP import policy, display the properties of the routes as they reside in the routing table with a `show route protocol rip` command.

Lab 4, Parts 1–3: RIP

- Configure and monitor RIP version 2.



Lab 4, Parts1–3: RIP

The slide shows the objectives for this lab.

Agenda: Routing Protocols and Policy

- Routing Table and Route Preferences
- Routing Policy
- J-Web Support for Routing Protocols and Policy
- Configuring and Monitoring Static Routing
- Interior Gateway Protocols
- Configuring and Monitoring RIP
- Configuring and Monitoring OSPF
- Configuring and Monitoring Basic BGP



Configuring and Monitoring OSPF

The slide highlights the topic we discuss next.

OSPF Protocol Overview

- OSPF is a link-state routing protocol
- OSPF reliably floods LSAs to distribute link-state information once an adjacency is formed
- Each router uses these LSAs to create a complete database for the network
- OSPF uses the SPF algorithm within the database to calculate the best route to every node in the network
- JUNOS software support for OSPF includes:
 - RFC 1587, *The OSPF NSSA Option*
 - RFC 2328, *OSPF Version 2*
 - RFC 2740, *OSPF for IPv6*
 - draft-katz-yeung-ospf-traffic-01.txt, *Traffic Engineering Extensions to OSPF*
 - draft-ietf-katz-ward-bfd-00.txt, *Bidirectional Forwarding Detection*



Link-State Protocol

OSPF is a link-state routing protocol designed for use within an AS. It is considered an IGP. Link-state protocols allow for faster reconvergence, support larger internetworks, and are less susceptible to bad routing information than distance-vector protocols.

LSA Flooding

Routers running OSPF send out information about their network links and the state of those links to other routers in the AS. This information is transmitted reliably to all other routers in the AS via LSAs. The other routers receive this information and store it locally on each router. This total set of information now contains all possible links in the network.

Continued on next page.

Link-State Database

In addition to flooding LSAs and discovering neighbors, a third major task of the link-state routing protocol is establishing the link-state database. The link-state (or topological) database stores the LSAs as a series of records. The important information for the shortest-path determination process is the advertising router's ID, its attached networks and neighboring routers, and the cost associated with those networks or neighbors.

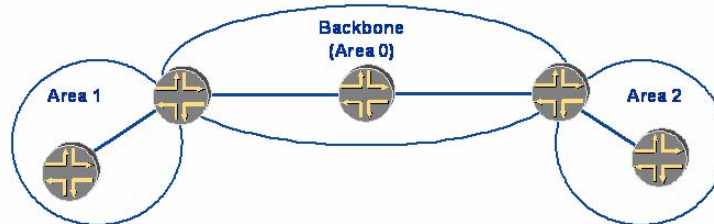
Shortest-Path-First Algorithm

OSPF uses the SPF algorithm or Dijkstra algorithm to calculate all at once the shortest paths to all destinations. It does this calculation by calculating a tree of shortest paths incrementally and picking the best candidate from that tree.

Standards

RFC 2328 defines OSPF version 2; RFC 1587 defines the OSPF NSSA option.

OSPF Areas (1 of 2)



■ Areas:

- Single AS can be divided into smaller groups called *areas*
- Reduces the link-state database because LSA flooding is now constrained to the area
- Routers maintain a separate link-state database on a per-area basis
- Each link-state database within an area still must be identical on all routers



Juniper *your* Net

6-51

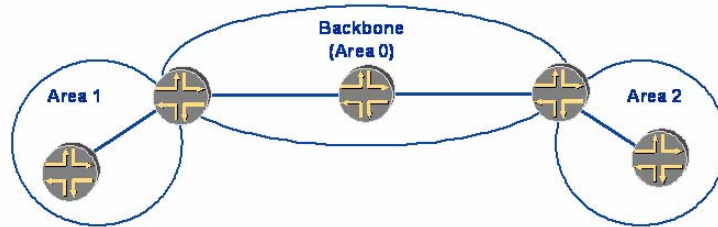
Copyright © 2007 Juniper Networks, Inc.

Proprietary and Confidential www.juniper.net

OSPF Areas

Using areas achieves the OSPF hierarchy. As mentioned previously, areas reduce the size of the link-state database on an individual router. Now, each router maintains a separate link-state database for each area to which it is connected.

OSPF Areas (2 of 2)



- Special OSPF area called the *backbone* area
 - Backbone area (0.0.0.0) distributes routing information between areas
 - All other OSPF areas must connect to the backbone area
 - All user traffic from one area to another must traverse the backbone

Backbone Area

To ensure correct routing knowledge and connectivity, OSPF maintains a special area called the backbone area. It is designated as Area 0. All other OSPF areas must connect themselves to the backbone for connectivity. All data traffic between OSPF areas must transit the backbone.

OSPF Router Terminology

- Internal router has all OSPF links in the same area
 - Within Area 0, also called a *backbone router*
- Backbone router
 - Any router with a link to Area 0
- ABRs
 - Routers that belong to more than one area are called *area border routers*
 - Connect OSPF areas to the backbone Area 0
- ASBRs
 - Routers that inject routing information from outside the OSPF domain are called *AS boundary routers*



Internal Routers

An OSPF router with all its links within an area is known as an internal router. If that router is located within the backbone area (0.0.0.0), it is also known as a backbone router.

Backbone Routers

Any OSPF router with a link to Area 0 (the backbone) is considered to be a backbone router. This router can also be an internal or area border router, depending on whether it has links to other, nonbackbone areas.

Area Border Routers

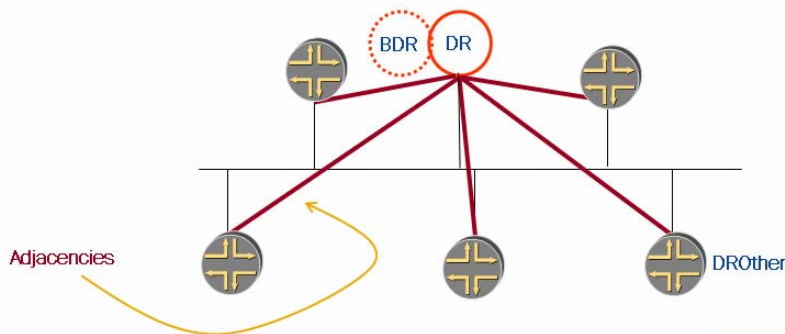
An OSPF router with links in two areas is called an area border router (ABR). The ABR is responsible for connecting OSPF areas to the backbone. It transmits network information between the backbone and the other areas.

Autonomous System Boundary Routers

An OSPF router that injects routing information from outside the OSPF AS is known as an autonomous system boundary router (ASBR). Typically, an ASBR is located in the backbone, but the OSPF specification allows an ASBR in other areas as well.

The Designated Router

- OSPF elects a designated router to represent a broadcast segment
 - Significantly reduces OSPF traffic on segment
 - A backup DR is also elected to recover for DR failures
 - DROther stations form adjacencies to the DR and BDR only

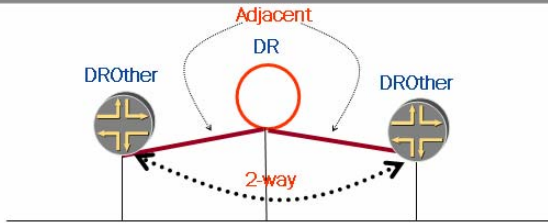


The Designated Router

OSPF routers want to form an adjacency with all routers with which they exchange hello packets. On a broadcast medium such as Ethernet, this desire can pose quite a problem. As more routers are added to the link, more adjacencies must be formed. This full-mesh requirement places extra load on the routers with little extra benefit because they all are advertising the same link information.

To avoid this problem, OSPF has a single router represent the broadcast link to the rest of the network. This router is called the designated router (DR). It is the DR's job to form an adjacency to all other routers on the link and to advertise the link-state information to the AS. A backup designated router (BDR) is also elected to take over in the event of a DR failure.

OSPF Neighbors Versus Adjacencies



```

user@host> show ospf neighbor extensive

```

Address	Intf	State	ID	Pri	Dead
172.16.30.254	fe-0/0/0.0	Full	10.250.240.8	128	30
area 0.0.0.5, opt 0x42, DR 172.16.30.254, BDR 172.16.30.253					
Up 00:10:50, adjacent 00:10:50					
172.16.30.253	fe-0/0/0.0	Full	10.250.240.35	128	30
area 0.0.0.5, opt 0x42, DR 172.16.30.254, BDR 172.16.30.253					
Up 00:10:50, adjacent 00:10:52					
172.16.30.252	fe-0/0/0.0	2Way	10.250.240.32	64	38
area 0.0.0.5, opt 0x42, DR 172.16.30.254, BDR 172.16.30.253					
Up 00:08:10					

2-way state to DROther routers is normal

Juniper your Net

Copyright © 2007 Juniper Networks, Inc. Proprietary and Confidential www.juniper.net

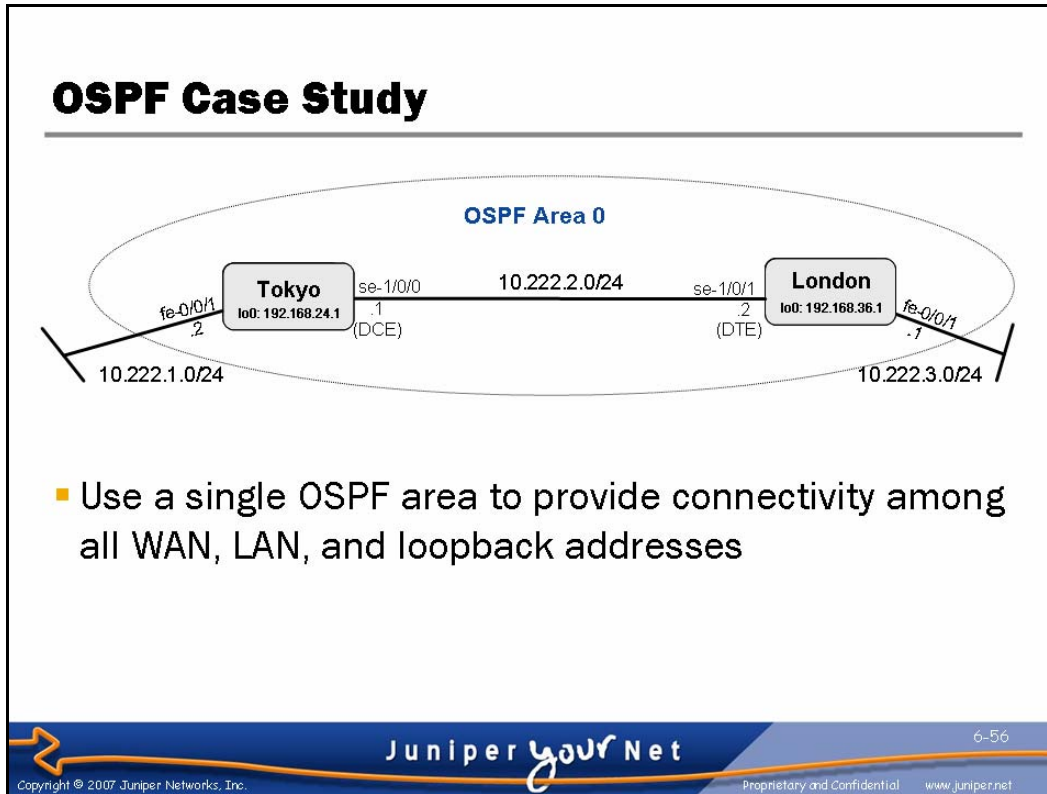
OSPF Neighbor Relationship

As soon as an OSPF router sees a hello packet on an interface, it starts to retain knowledge of that neighbor. You can display this information with the operational CLI command **show ospf neighbor**.

On the slide, this router has three neighbors on the `fe-0/0/0.0` interface. Two of the three routers are the DR and the BDR; full adjacencies exist with them. Each of the hello packets received from all three routers lists their addresses.

The router that is in a two-way state is a neighbor on the link, but it is not the DR/BDR. This router reaches the two-way state because the DR and BDR can see its hello packets, and this router's own RID is located in the received hello. For broadcast media, it is acceptable to have some neighbors in the two-way state.

The address column is the interface IP address of the neighboring router. The ID column is the RID of the neighboring router.

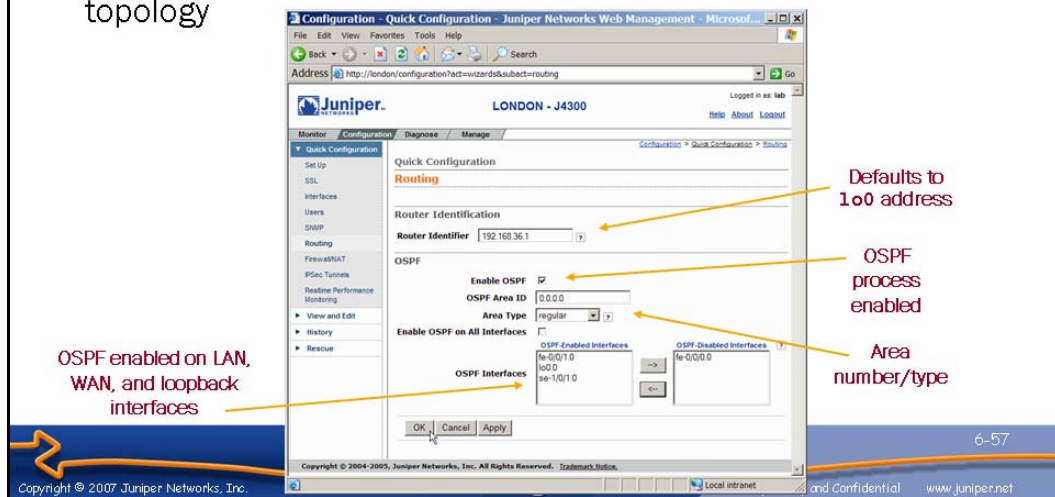


OSPF Example

We use the topology shown on the slide to demonstrate single-area OSPF routing. We will enable OSPF on the WAN, LAN, and loopback interfaces of both *Tokyo* and *London*. This setup will provide connectivity between all networks in the topology.

Configuring OSPF: J-Web

- Use the J-Web OSPF wizard at the Configuration > Quick Configuration > Routing and Protocols page
 - Configuration goal: A single-area OSPF network using the sample topology



J-Web OSPF Routing Wizard

The J-Web Configuration > Quick Configuration > Routing and Protocols > OSPF Routing wizard greatly simplifies configuration of the OSPF protocol. For single-area configurations you simply check the Enable OSPF box and select the logical interfaces which should participate in the protocol. Like all wizards, clicking OK or Apply commits your changes.

J-Web populates the Router Identifier (RID) with the router's loopback address. This explicit RID configuration requires you to run OSPF on the 100.0 interface to advertise the loopback interface's address into OSPF. If the RID is not explicitly configured, the router defaults to using the 100.0 address as the RID and automatically advertises the loopback interface's address into OSPF.

The Resulting OSPF Configuration

```
[edit]
lab@London# show routing-options
router-id 192.168.36.1;
```

```
[edit]
lab@London# show protocols ospf
area 0.0.0.0 {
  interface fe-0/0/1.0;
  interface se-1/0/1.0;
  interface lo0.0;
}
```

Required due to explicit RID;
can be set to passive

- Default OSPF import and export policies are in effect
 - Explicit declaration of lo0-based RID requires that OSPF run on loopback interface
 - Otherwise, the loopback route will not be advertised



The Results

The configuration that results from the J-Web OSPF wizard is piped through **display set** to reveal the corresponding CLI syntax:

```
[edit]
lab@London# show routing-options | display set
set routing-options router-id 192.168.36.1
```

```
[edit]
lab@London# show protocols ospf | display set
set protocols ospf area 0.0.0.0 interface fe-0/0/1.0
set protocols ospf area 0.0.0.0 interface se-1/0/1.0
set protocols ospf area 0.0.0.0 interface lo0.0
```

Monitoring OSPF: J-Web

- Use the J-Web Monitor > Routing > OSPF Information page to monitor general OSPF operation

The screenshot shows the Juniper J-Web interface for monitoring OSPF. The left sidebar contains navigation links for System, Chassis, Interfaces, Routing, and various protocols. The main content area is titled 'OSPF Information' and includes three sections:

- OSPF Neighbors:** A table showing neighbor details. An annotation points to this section with the text 'OSPF neighbor and interface status'.
- OSPF Interfaces:** A table showing interface details. An annotation points to this section with the text 'OSPF neighbor and interface status'.
- OSPF Statistics:** A table showing protocol statistics. An annotation points to this section with the text 'OSPF protocol statistics'.

At the bottom of the page, there is a footer with the Juniper logo and the text 'Juniper your Net'.

Monitoring OSPF with J-Web

The J-Web Monitor > Routing > OSPF Information page allows you to monitor general OSPF operation. It details overall OSPF statistics as well as neighbor and interface status. Clicking an interface reveals OSPF configuration for that interface.

Monitoring OSPF Using the CLI (1 of 5)

- Use the `show ospf route` command to display routes learned and advertised into OSPF
 - Includes routes for interfaces running OSPF

Use switches to filter by
OSPF route (LSA) type

```
lab@London> show ospf route ?
Possible completions:
<[Enter]>      Execute this command
abr            Display OSPF routes to area border routers
asbr          Display OSPF routes to AS border routers
detail        Display detailed output
extern        Display external OSPF routes
instance      Name of OSPF instance
inter         Display interarea OSPF routes
intra         Display intraarea OSPF routes
|            Pipe through a command

lab@London> show ospf route detail
Prefix      Path  Route      NH  Metric  NextHop      Nexthop
Type        Type      Type      Interface
192.168.24.1 Intra Router    IP   12      se-1/0/1.0
area 0.0.0.0, origin 192.168.24.1 optional-capability 0x0,
10.222.1.0/24 Intra Network IP   13      se-1/0/1.0
area 0.0.0.0, origin 192.168.24.1
. . .
```



Displaying OSPF Route Information

The `show ospf route` command displays those routes in the unicast routing table, `inet.0`, that were installed by OSPF. The use of additional keywords, such as `abr`, allows you to display only OSPF routes learned by specific LSA types. The output fields of the `show ospf route` command are the following:

- Prefix: Displays the destination of the route.
- Route/Path Type: Displays how the route was learned:
 - ABR: Route to area border router;
 - ASBR: Route to AS border router;
 - Ext: External router;
 - Inter: Interarea route;
 - Intra: Intra-area route; or
 - Network: Network route.
- Metric: Displays the route's metric value.
- Next hop i/f: Displays the interface through which the route's next hop is reachable.

Continued on next page.

Displaying OSPF Route Information (contd.)

- `Next hop addr`: Displays the address of the next hop.
- `area` (detailed output only): Displays the area ID of the route.
- `options` (detailed output only): Displays the option bits from the LSA.
- `origin` (detailed output only): Displays the router from which the route was learned.

Monitoring OSPF Using the CLI (2 of 5)

- Use the `show ospf interface` command to display the OSPF interface parameters
 - Add the **detail** or **extensive** switches for additional information

```
lab@London> show ospf interface
```

Interface	State	Area	DR ID	BDR ID	Nbrs
fe-0/0/1.0	DR	0.0.0.0	192.168.36.1	0.0.0.0	0
lo0.0	DR	0.0.0.0	192.168.36.1	0.0.0.0	0
se-1/0/1.0	PtToPt	0.0.0.0	0.0.0.0	0.0.0.0	1

DR/BDR not elected on point-to-point links

An OSPF neighbor was detected

Displaying OSPF Interface Parameters

The `show ospf interface` command displays information relating to the interfaces on which the OSPF protocol is configured to run. For OSPF, the output fields are the following:

- **Interface:** Displays the name of the interface running OSPF.
- **State:** Displays the state of the interface. It can be BDR, Down, DR, DROther, Loop, PtToPt, or Waiting.
- **Area:** Displays the number of the area in which the interface is located.
- **DR ID:** Displays the address of the area's DR.
- **BDR ID:** Displays the BDR for a particular subnet.
- **Nbrs:** Displays the number of neighbors on this interface.
- **Type (detail and extensive output only):** Displays the type of interface. It can be LAN, NBMA, P2MP, P2P, or Virtual.
- **Address (detail and extensive output only):** Displays the IP address of the neighbor.
- **Mask (detailed and extensive output only):** Displays the mask of the interface.
- **MTU (detailed and extensive output only):** Displays the interface's MTU.
- **Cost (detail and extensive output only):** Displays the interface's cost (metric).

Continued on next page.

Displaying OSPF Interface Parameters (contd.)

- `DR addr` (detailed and extensive output only): Displays the address of the DR.
- `BDR addr`: Displays the address of the BDR.
- `Adj count` (detailed and extensive output only): Displays the number of adjacent neighbors.
- `Flood list` (extensive output only): Displays the list of LSAs pending flood on this interface.
- `Ack list` (extensive output only): Displays the list of pending acknowledgments on this interface.
- `Descriptor list` (extensive output only): Displays the list of packet descriptors.
- `Dead` (detailed and extensive output only): Displays the configured value for the dead timer.
- `Hello` (detailed and extensive output only): Displays the configured value for the hello timer.
- `ReXmit` (detailed and extensive output only): Displays the configured value for the retransmit timer.
- `OSPF area type` (detailed and extensive output only): Displays the type of OSPF area, which can be `Stub`, `Not Stub`, or `NSSA`.

Monitoring OSPF Using the CLI (3 of 5)

- Use the `show ospf neighbor` command to display adjacency information

```
lab@London> show ospf neighbor
```

Address	Interface	State	ID	Pri	Dead
10.222.2.1	se-1/0/1.0	Full	192.168.24.1	128	38

- Clear adjacencies with the `clear ospf neighbor` command

```
lab@London> clear ospf neighbor
```

```
lab@London> show ospf neighbor
```

Address	Interface	State	ID	Pri	Dead
10.222.2.1	se-1/0/1.0	ExStart	192.168.24.1	128	38



Displaying Adjacency Information

The `show ospf neighbor` command displays OSPF adjacency status. The output fields include the following:

- Address: Displays the address of the neighbor.
- Intf: Displays the interface through which the neighbor is reachable.
- State: Displays the state of the neighbor, which can be Attempt, Down, Exchange, ExStart, Full, Init, Loading, or 2Way.
- ID: Displays the RID of the neighbor.
- Pri: Displays the priority of the neighbor to become the DR.
- Dead: Displays the number of seconds until the neighbor becomes unreachable.
- area (detailed and extensive output only): Displays the area in which the neighbor is located.
- opt (detailed and extensive output only): Displays the option bits from the neighbor.
- DR (detailed and extensive output only): Displays the address of the DR.
- BDR (detailed and extensive output only): Displays the address of the BDR.

Continued on next page.

Displaying Adjacency Information (contd.)

- `Up` (detailed and extensive output only): Displays the length of time since the neighbor came up.
- `adjacent` (detailed and extensive output only): Displays the length of time since the adjacency with the neighbor was established.

Use the **`clear ospf neighbor`** command to clear an OSPF adjacency. Note that in most cases the adjacency should be reformed immediately.

Monitoring OSPF Using the CLI (4 of 5)

- Use the **show ospf database** command to display entries in the link-state database
 - Filter the display by LSA type
 - Use the **detail** or **extensive** switches for added information

```
lab@London> show ospf database ?
Possible completions:
<[Enter]>          Execute this command
advertising-router Router ID of advertising router
area              OSPF area ID
asbrsummary       Show summary AS boundary router link-state database
brief            Display brief output (default)
detail           Display detailed output
extensive        Display extensive output
extern           Show external link-state database
. . .
```

```
lab@London> show ospf database

    OSPF link state database, area 0.0.0.0
Type      ID          Adv Rtr      Seq         Age    Opt   Cksum  Len
Router    192.168.24.1    192.168.24.1 0x800000005 1375   0x2    0xce62  72
Router    *192.168.36.1   192.168.36.1 0x800000006 1386   0x2    0x9b79  72
```

Self originated LSAs marked with *



Displaying OSPF Database Entries

The **show ospf database** command displays entries in the protocol's link-state database. The display is organized by LSA types. The **show ospf database** command options include:

- **brief** (optional): Displays a brief listing of all entries in the OSPF link-state database. This is the default setting.
- **detail** (optional): Displays detailed information about the entries in the OSPF link-state database.
- **extensive** (optional): Displays extremely detailed information about the entries in the OSPF link-state database.

Continued on next page.

Displaying OSPF Database Entries (contd.)

- **LSA filters** (optional): Displays one or more of the following LSA filters. If you specify more than one filter, only LSAs that match all the filters are displayed. For example, the command **show ospf database detail router lsa-id 10.0.0.1** displays all router LSAs in all areas that have an LSA identifier of 10.0.0.1. The filters are the following:
 - **advertising-router address**: Displays the LSAs advertised by a particular router.
 - **area area-id**: Displays the LSAs in a particular area.
 - **lsa-id lsa-id** (optional): Displays the LSA with the specified LSA identifier.
 - **lsa-type**: Displays specific types of LSAs. You can specify **asbrsummary**, **extern**, **netsummary**, **network**, **nssa**, or **router**.
 - **summary** (optional): Displays summary information about the OSPF link-state database.

Monitoring OSPF Using the CLI (5 of 5)

- Use the `clear ospf database` command to clear the link-state database
 - Normally, existing LSAs are simply reflooded over existing adjacencies
 - OSPF supports a **purge** option that forces refresh of all LSAs

```
lab@London> clear ospf database purge
```

```
lab@London> show ospf database
```

```

  OSPF link state database, area 0.0.0.0
  Type      ID                Adv Rtr          Seq            Age      Opt  Cksum  Len
  Router    192.168.24.1         192.168.24.1    0x800000008    3600    0x2   0xc865  72
  Router    *192.168.36.1           192.168.36.1    0x80000000a    0       0x2   0x937d  72

```

The **purge** switch forces all LSAs to the maximum age; the originating router will refresh the LSA if it is still valid

Clearing Database Entries

The `clear ospf database` command clears entries from the link-state database. After the command is entered, the router begins the database synchronization process with its neighboring routers such that, in most cases, the database returns to its prior state.

The `clear ospf database` command supports an optional **purge** switch. By including the purge switch, you force the local router to set *all* LSAs in its database to the maximum age. These LSAs are then reflooded according to the OSPF specification, which states that a router must regenerate any LSA that it has set to maximum age, regardless of whether the LSA was generated by the local router. All routers receive the newly flooded maximum age LSAs; the router that originated a given LSA is forced to refresh that LSA when it receives a copy of that LSA with an indication that it has reached the maximum age.

Albeit somewhat disruptive, this procedure tends to eliminate stale or bogus database entries without having to wait for the normal aging-out process, which can take as long as 3600 seconds (one hour). Note that other vendors' OSPF implementations might not be prepared for a simultaneous reflooding of every LSA in the network or for another router to increase the age of LSAs that their routers originated. Therefore, you should not use this feature in a production network without prior interoperability testing.

Configuring IGP Tracing

- Use tracing to debug the operation of your IGP

- A typical OSPF tracing configuration:

```
[edit protocols ospf]
lab@London# show traceoptions
file ospf-trace;
flag error detail;
flag hello detail;
flag lsa-update;
```

- Monitor the resulting *ospf-trace* log file using the **monitor start log-file-name** or the **show log log-file-name** CLI commands
 - Use **Esc-q** to toggle terminal output when monitoring
- Turn off tracing by deleting the *traceoptions* stanza

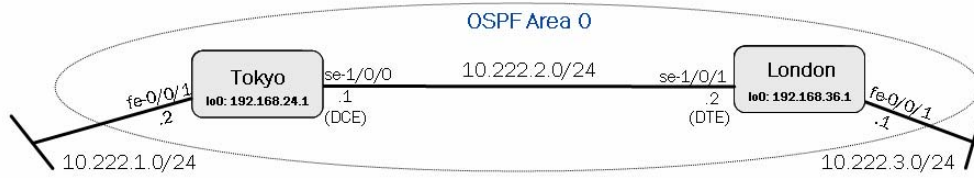


IGP Tracing

To perform debugging functions on the OSPF routing process, use the JUNOS software **traceoptions** function. The trace output (debug information) is directed to the named log file, which is stored in the */var/log* directory on the router's primary compact flash drive. You can view the log file using the **monitor start** or **show log** operational-mode commands. In addition to specifying the trace file, you also must tell the router what information you want to trace. You can accomplish this specifying one or more **flag** keywords.

While you can only direct tracing to a single file, you can trace many options by using the **flag** keyword multiple times. In addition, you can add granularity by using the **detail**, **receive**, and **send** flag modifiers.

IGP Troubleshooting Case Study (1 of 2)

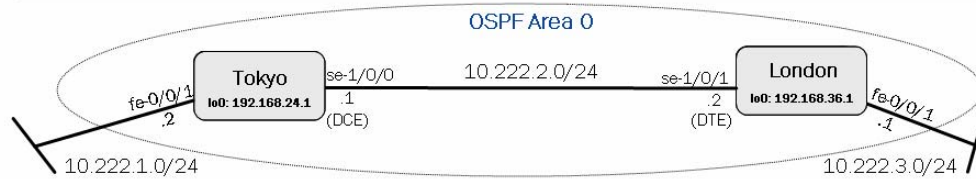


- An OSPF adjacency will not form across the WAN link; ping testing of the WAN succeeds

Problem

An OSPF adjacency will not form across the WAN link, but ping testing of the link is successful. See the next page for a solution to this problem.

IGP Troubleshooting Case Study (2 of 2)



- **Solution: Trace OSPF to see what is happening!**
 - Can you spot the problem based on the sample trace output?

```
lab@London> monitor start ospf-trace
```

```
lab@London>
```

```
*** ospf-trace ***
```

```
Jul 30 16:39:42 OSPF periodic xmit from (null) to 224.0.0.5 (IFL 71)
Jul 30 16:39:48 OSPF packet ignored: authentication type mismatch (0)
    from 10.222.2.1
Jul 30 16:39:48 OSPF periodic xmit from 10.222.29.1 to 224.0.0.5 (IFL 70)
Jul 30 16:39:52 OSPF periodic xmit from (null) to 224.0.0.5 (IFL 71)
Jul 30 16:39:56 OSPF packet ignored: authentication type mismatch (0)
    from 10.222.2.1
```



Answer

The OSPF authentication configuration does not match between the two routers as shown by the authentication type mismatch messages in the *ospf-trace* log file.

Lab 4, Parts 4–5: OSPF

- Configure and monitor single-area OSPF using J-Web.
- Configure and monitor multiarea OSPF using the CLI (optional).



Lab 4, Parts 4–5: OSPF

The slide shows the objectives for this lab.

Agenda: Routing Protocols and Policy

- Routing Table and Route Preferences
- Routing Policy
- J-Web Support for Routing Protocols and Policy
- Configuring and Monitoring Static Routing
- Interior Gateway Protocols
- Configuring and Monitoring RIP
- Configuring and Monitoring OSPF
- Configuring and Monitoring Basic BGP



Configuring and Monitoring Basic BGP

This slide highlights the topic we discuss next.

What Is BGP?

Each AS is under separate administrative control

Interdomain Routing

Autonomous Network A Autonomous Network B

- **BGP 4:**
 - Is an interdomain routing protocol
 - Supports CIDR and route attributes that accommodate complex routing policy
 - Is a *path-vector* protocol that uses incremental updates and reliable TCP transport
 - Views the Internet as a collection of ASs
 - Normally requires explicitly defined *peers* for added security and control
 - Is an IETF standard defined in RFC 4271 (supersedes RFC 1771)

3-74

Copyright © 2006 Juniper Networks, Inc. Proprietary and Confidential www.juniper.net

What Is BGP?

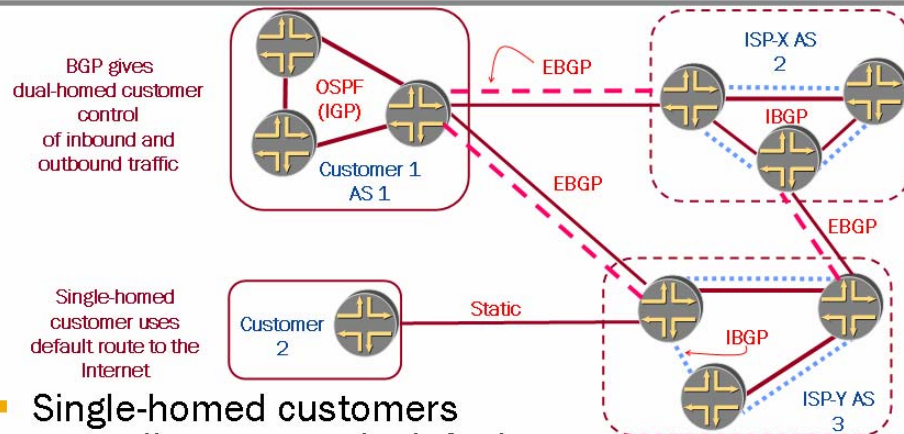
BGP is an interautonomous system (inter-AS) routing protocol and is sometimes called a *path-vector routing protocol* because it uses an AS path, used as a vector, to prevent interdomain routing loops. The term *path vector*, in relation to BGP, means that BGP routing information includes a series of AS numbers, indicating the path that a route takes through the network.

BGP exchanges routing information among ASs or domains. An AS is a set of routers that operate under the same administration. BGP routing information includes the complete route to each destination. BGP uses the routing information to maintain a database of network layer reachability information (NLRI), which it exchanges with other BGP systems. BGP uses the NLRI to construct a graph of AS connectivity, thus allowing BGP to remove routing loops and enforce policy decisions at the AS level.

BGP is a classless routing protocol, which supports prefix routing, regardless of the class definitions of IPv4 addresses. BGP routers exchange routing information between peers. The peers must be connected directly for inter-AS BGP routing (unless certain configuration changes are done).

BGP version 4 (BGP4) is essentially the only exterior gateway protocol (EGP) currently used in the Internet. It is defined in RFC 4271, which made the former standard of more than 10 years, RFC 1771, obsolete.

Interdomain Routing Scenarios



- Single-homed customers normally use a static default route
- Multihomed customers benefit from BGP route selection intelligence and policy controls

Single-Homed Networks

Networks with a single upstream connection receive little benefit from running a dynamic routing protocol with their service provider. These customers typically use a static default route to send all external traffic towards the Internet.

Multihomed Networks

BGP is normally used when a network has multiple upstream connections. BGP's policy controls provide the ability to optimize inbound and outbound traffic flows based on a network's technical and business constraints. Like all routing protocols, BGP can also dynamically detect and route around link and node failures.

BGP Peering

- BGP sessions are normally established between explicitly defined peers for added control
- Two types of peering sessions:
 - EBGP (external) peers with different ASs
 - Usually to physical address
 - IBGP (internal) peers within the same AS
 - Full-mesh requirement

Juniper your Net

6-76

Copyright © 2007 Juniper Networks, Inc. Proprietary and Confidential www.juniper.net

BGP Peers

BGP is a protocol in which routing information exchanges occur between exactly two nodes, called peers. These peers can be connected either directly or remotely.

EBGP Versus IBGP

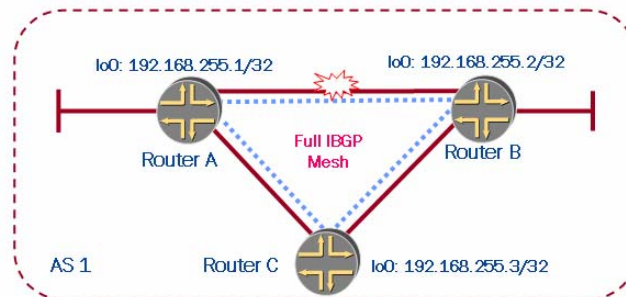
BGP supports two different types of exchanges of routing information. Exchanges between ASs are called *external BGP*, or *EBGP*, sessions and handle inter-AS routing. Exchanges within an AS are called *internal BGP*, or *IBGP*, sessions, and handle intra-AS routing.

An EBGP peer connection is between a device in one AS and another device in a different AS. The connection between the two ASs consists of a physical connection and a BGP connection. The physical connection is a shared data link layer subnetwork between the two ASs. On this shared subnetwork, each AS has at least one border gateway belonging to that AS. The BGP connection exists between BGP speakers in each of the ASs. This session can communicate destinations that can be reached through the advertising AS. The EBGP connection typically is established between immediately connected devices located in two different ASs because the time-to-live (TTL) value of EBGP packets is equal to 1, by default.

An IBGP connection is established between all BGP speaking routers within an AS. To avoid intra-AS loops the BGP protocol specifies that IBGP learned routes are not to be advertised over IBGP sessions. This requires a full-mesh of BGP speaking routers within an AS. Route reflection and confederations are advanced BGP features that loosen this requirement for large ASs.

IBGP and Loopback Peering

- Best practice is to peer between loopback interfaces
 - More stable
 - Not tied to a single physical path
- The AS needs an IGP so that IBGP speakers can reach each others' loopback addresses



IBGP Use of Loopback Interfaces

IBGP peers often use loopback interfaces. The advantage of using loopback interfaces is that they eliminate a dependency that would otherwise occur when you use the IP address of a physical interface to configure BGP. The slide shows a network in which using the loopback interface is advantageous.

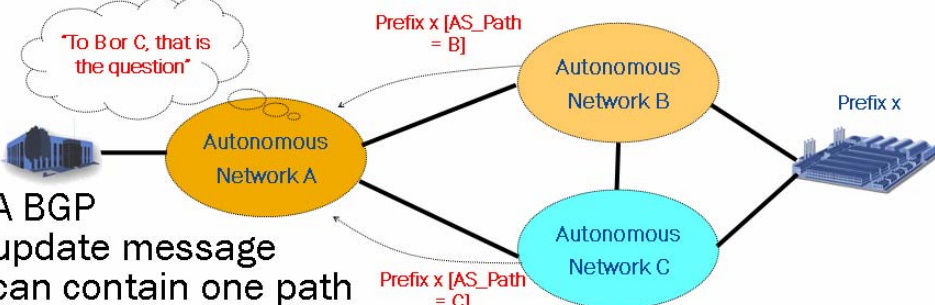
On the slide, Routers A, B, and C run IBGP within AS 1. If Router A were to specify the IP address of an Ethernet interface on Router B in the remote neighbor with the router configuration command, and if the specified interface were to become unavailable, Router A would not be able to establish a TCP connection with router B. Instead, Router A specifies the IP address of the loopback interface that Router B defines. When the loopback interface is used, BGP does not have to rely on the availability of a particular interface for making TCP connections. Router A specifies the IP address of the loopback interface (192.168.255.2) of Router B. If the direct link between Router A and Router B fails, their IBGP session will be routed through Router C.

Note that BGP rarely uses loopback interfaces between EBGp peers because EBGp peers usually are connected directly. EBGp peers therefore depend on a particular physical interface for connectivity (however, exceptions include parallel paths).

IGP Requirement

The AS needs IGP or static routes so that IBGP speakers can establish TCP sessions to each others' loopback interfaces. BGP's TCP session is established using regular routing tables. Internal peers can be anywhere in the AS and need not be directly connected to each other, as shown on the slide.

BGP Updates and Attributes

- 
- A BGP update message can contain one path advertisement and its associated attributes
 - Multiple prefixes can be advertised in one update if they share the same BGP attributes
 - Can also list one or more routes that are no longer reachable
 - BGP compares the AS path and other attributes to select the best path
 - Accommodates administrative control over route selection

BGP Updates

Routes in BGP consist of destination networks and attributes associated with those routes. Each BGP update contains one path advertisement. However, many destinations can share the same path. The receiving device assumes that the route remains active (should the BGP next hop be accessible) until the originator explicitly withdraws it or until the session is terminated.

Once a connection is open and active, BGP sends routes. BGP routes consist of destination prefixes, each associated with BGP attributes. (For IGPs, *metrics* is the term used to describe their attributes.) Some of the complexities of BGP are the variety of these metrics (or *attributes*), the order of their execution, and various rules that can be applied to the attributes.

Continued on next page.

BGP Attributes

Both you and BGP itself can associate one or more attribute with a route advertisement. Attributes carry descriptive information about the route and are used in choosing the best path to a destination.

BGP attributes describe the following:

- The next hop for a packet sent to a particular destination;
- Various numeric-type attributes;
- The path through ASs that a routing announcement has traversed to arrive at the destination where it is now; and
- The method of generation for the prefix, or which protocol originated the route.

BGP Active Route Selection Summary

■ Selection summary:

1. Can the BGP next hop be resolved?
2. Prefer the highest local-preference value
3. Prefer the shortest AS-path length
4. Prefer the lowest origin value
5. Prefer the lowest MED value
6. Prefer routes learned using EBGP over routes learned using IBGP
7. Prefer routes with the lowest IGP metric
8. Prefer paths with the shortest cluster length
9. For EBGP-received routes, prefer the current active route; otherwise, prefer routes from the peer with the lowest RID
10. Prefer routes from the peer with the lowest peer ID



Summary of BGP Active Route Selection

When a BGP router learns the same route from multiple BGP speakers, it must decide which route to install into the routing table as an active route. The following steps provide a summary overview of the BGP active route selection algorithm. Note that some details are omitted here in the interest of brevity:

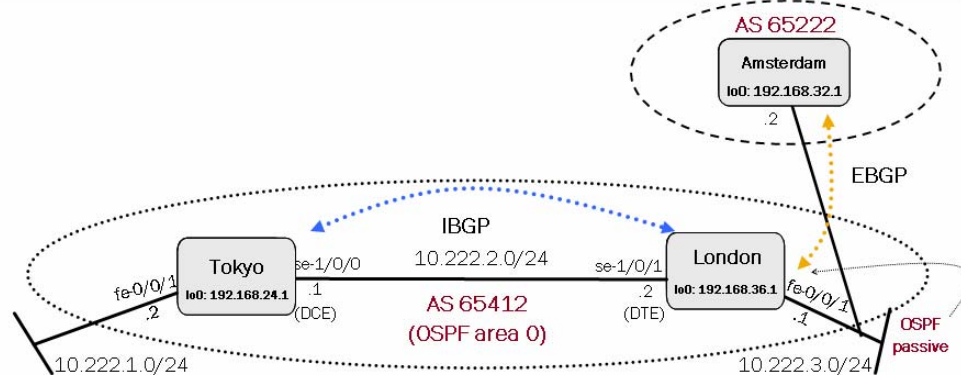
1. The router first must verify that it has a current route in the `inet.0` routing table to the IP address in the BGP next-hop attribute field.
2. The router then compares routes for the *highest* local preference (the only choice based on a higher, rather than lower, value).
3. The router evaluates the AS-path attribute next, where a shorter path is preferred. This attribute is often a common tiebreaker for routes.
4. The router evaluates the origin code. The lowest origin code is preferred.
5. If any of the remaining routes are advertised from the same neighboring AS, the router checks the multiple exit discriminator (MED) attributes for a lowest value. The absence of a MED value is interpreted as a MED of 0.

Continued on next page.

Summary of BGP Active Route Selection (contd.)

6. If multiple routes remain, the router prefers any routes learned via an EBGp peer over routes learned via an IBGP peer. If all remaining routes were learned through EBGp, the router skips to Step 9.
7. If the remaining routes were learned through IBGP, the router uses the path with the lowest IGP cost to the IBGP peer.
8. The router then examines the cluster-list attribute for the shortest length. The cluster list is similar in function to an AS path.
9. The router prefers the route advertised from the peer with the lowest router ID. However, for EBGp-received routes only, the router prefers the current active route when comparing routes received from different neighboring ASs.
10. The router prefers routes from the router with the lowest peer ID.

Sample BGP Topology—IBGP

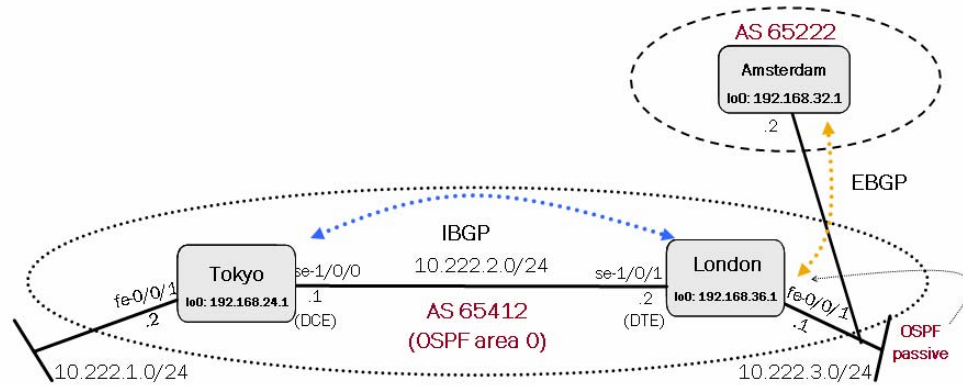


- *Tokyo* and *London* are in OSPF Area 0 and peer between loopback addresses for IBGP
 - *London's* fe-0/0/1 interface is set to passive to prevent IGP adjacency formation

IBGP Example

We use the topology shown on the slide to demonstrate both IBGP and EBGP routing. *Tokyo* and *London* are in the same AS and will have an IBGP session between loopback addresses. We will enable OSPF on the WAN, LAN, and loopback interfaces of *Tokyo* and *London* to provide the connectivity required for IBGP peering between loopback addresses.

Sample BGP Topology—EBGP



- EBGp peering to physical addresses
 - Policy required on *London* and *Amsterdam* to advertise direct and OSPF routes into BGP

EBGP Example

We will establish an EBGp session between *Amsterdam* and *London*. We will also configure the *London* and *Amsterdam* routers with a policy that injects direct and OSPF routes into BGP. This setup will provide connectivity between all networks in the topology.

J-Web BGP Configuration: EBGp

- Use the J-Web BGP wizard at the Configuration > Quick Configuration > Routing and Protocols page
 - CLI or J-Web-based edits are needed for the IBGP session and export policy needed in this example

The screenshot shows the Juniper J-Web Configuration interface for router LONDON - J4300. The 'Routing' section is active, and the 'BGP' configuration page is displayed. The following fields are visible and annotated:

- Router Identification:** Router Identifier is set to 192.168.36.1. An annotation states: "Router ID defaults to 100 address".
- BGP:** The 'Enable BGP' checkbox is checked. An annotation states: "BGP is enabled".
- Autonomous System Number:** Set to 64512.
- Peer Autonomous System Number:** Set to 65222. An annotation states: "Local and remote AS numbers".
- Peer Address:** Set to 10.222.3.2.
- Local Address:** Set to 10.222.3.2. An annotation states: "Peering address, local address not needed for physical peering".

The interface also shows a sidebar with navigation options like 'Quick Configuration', 'Set up', 'SSL', 'Interfaces', 'Users', 'SNMP', 'Routing', 'Firewall/FAT', 'IPSec Tunnels', 'Realtime Performance Monitoring', 'View and Edit', 'History', and 'Rescue'. The bottom of the page includes copyright information for Juniper Networks, Inc. and a 'Local Intranet' link.

J-Web BGP Configuration

You can use the J-Web Configuration > Quick Configuration > Routing and Protocols > BGP Routing page to configure a single BGP peer without policy. More advanced BGP configurations require using the CLI or J-Web's View and Edit functionality. The slide demonstrates *London's* EBGp session to *Amsterdam*.

Configuring a BGP session requires a local and a remote AS number as well as local and remote peering IP addresses. The local IP address defaults to the IP address of the next-hop interface used to reach the remote IP address. For directly connected EBGp neighbors, this is the desired behavior, and you do not have to explicitly configure the address in J-Web. For IBGP sessions between loopback addresses, you must configure the loopback IP address as the local IP address.

The Initial EBGW Configuration

- The J-Web BGP wizard should get your BGP session established
 - The initial EBGW configuration requires tuning for EBGW export policy and IBGP peering session to *Tokyo*

```
[edit]
lab@London# show routing-options
router-id 192.168.36.1;
autonomous-system 65412;

[edit]
lab@London# show protocols bgp
group jweb-bgp {
    peer-as 65222;
    neighbor 10.222.3.2;
}
```



Resulting J-Web Configuration

This slide shows the configuration produced by using the J-Web BGP wizard. *London* still requires an IBGP session and export policy to be configured using the CLI or J-Web View and Edit functionality.

The Modified BGP Configuration

- The definition of *London's* IBGP session and EBGP export policy are displayed with the `compare` function

```
lab@London# show | compare
[edit protocols bgp group jweb-bgp]
+ export direct-and-ospf-to-bgp;
[edit protocols bgp]
group jweb-bgp { ... }
group internal {
+   type internal;
+   local-address 192.168.36.1;
+   neighbor 192.168.24.1;
+ }
[edit]
+ policy-options {
+   policy-statement direct-and-ospf-to-bgp {
+     term term1 {
+       from protocol ospf;
+       then accept;
+     }
+     term term2 {
+       from {
+         protocol direct;
+         interface [ lo0.0 se-1/0/1.0 fe-0/0/1.0 ];
+       }
+       then accept;
+     }
+   }
+ }
```

direct-and-ospf-to-bgp policy is applied to EBGP peer

Internal peer group, note loopback-based peering

The policy accepts OSPF and selected direct routes

The Results

The following capture reveals the CLI syntax needed to create the BGP configuration shown on the slide:

```
[edit]
lab@London# show policy-options | display set
set policy-options policy-statement direct-and-ospf-to-bgp term term1 from
protocol ospf
set policy-options policy-statement direct-and-ospf-to-bgp term term1 then
accept
set policy-options policy-statement direct-and-ospf-to-bgp term term2 from
protocol direct
set policy-options policy-statement direct-and-ospf-to-bgp term term2 from
interface lo0.0
set policy-options policy-statement direct-and-ospf-to-bgp term term2 from
interface se-1/0/1.0
set policy-options policy-statement direct-and-ospf-to-bgp term term2 from
interface fe-0/0/1.0
set policy-options policy-statement direct-and-ospf-to-bgp term term2 then
accept
```

Continued on next page.

The Results (contd.)

```
[edit]
lab@London# show protocols bgp | display set
set protocols bgp group jweb-bgp export direct-and-ospf-to-bgp
set protocols bgp group jweb-bgp peer-as 65222
set protocols bgp group jweb-bgp neighbor 10.222.3.2
set protocols bgp group internal type internal
set protocols bgp group internal local-address 192.168.36.1
set protocols bgp group internal neighbor 192.168.24.1
```

Monitoring BGP: J-Web

- Use the J-Web Monitor > Routing > BGP Information page to monitor general BGP operation
 - All BGP neighbors should be in the Established state

Peer summary

Neighbor summary

Both BGP sessions are correctly established!

Monitoring BGP with J-Web

The J-Web Monitor > Routing > BGP Information page displays the number of prefixes learned from BGP and status information for each BGP neighbor. The possible states for BGP neighbors are the following:

- *Idle*: This is the first stage of a connection. BGP is waiting for a start event.
- *Active*: BGP is initiating a transport protocol connection in an attempt to connect to a peer. If the connection is successful, BGP sends an open message.
- *Connect*: BGP is waiting for the transport protocol connection to complete.
- *OpenSent*: BGP has sent an open message and is waiting to receive an open message from the peer.
- *OpenConfirm*: BGP has acknowledged receipt of an open message from the peer and is waiting to receive a keepalive or notification message.
- *Established*: The BGP session is established, and the peers are exchanging update messages.

Monitoring BGP Operation: CLI

■ Useful CLI commands for BGP monitoring:

```
show bgp summary
show bgp neighbor
show route protocol bgp
clear bgp neighbor neighbor-address
show route advertising-protocol neighbor-address
show route receive-protocol bgp neighbor-address
```

BGP routing summary

```
lab@London> show bgp summary
Groups: 2 Peers: 2 Down peers: 0
Table Tot Paths Act Paths Suppressed History Damp State Pending
inet.0 2 1 0 0 0 0 0
Peer AS InPkt OutPkt OutQ Flaps Last Up/Dwn State|#Active/Received/Damped...
```

Peer	AS	InPkt	OutPkt	OutQ	Flaps	Last Up/Dwn	State	#Active/Received/Damped...
192.168.24.1	64512	840	841	0	0	6:58:54	0/0/0	0/0/0
10.222.3.2	65222	846	849	0	0	7:02:09	1/2/0	0/0/0

Received, active, and damped routes

London has two BGP sessions established


Juniper your Net

Copyright © 2007 Juniper Networks, Inc. Proprietary and Confidential www.juniper.net

Monitoring BGP Operation

JUNOS software has a wide variety of BGP monitoring commands. This slide displays the following commands:

- **show bgp summary:** Displays the overall BGP information, including the state of BGP peer session establishment.
- **show bgp neighbor:** Shows the BGP neighbor database.
- **show route protocol bgp:** Shows the BGP learned routes in the routing table.
- **clear bgp neighbor:** Performs a soft or hard reset of the session to the specified BGP neighbor.
- **show route advertising-protocol bgp:** Displays the routes being sent to the specified BGP neighbor. The output shows routes after any BGP export policies are applied.
- **show route receive-protocol bgp:** Displays the routes being received from the specified BGP neighbor. The output show routes before BGP import policies (except for route-filters) are applied.

Displaying BGP Routes

```
lab@London> show route protocol bgp ?
Possible completions:
<[Enter]>          Execute this command
<destination>      IP address and optional prefix length of destination
advertising-protocol Show information in format intended for particular
routing protocol
all                Show all entries, including hidden entries
aspath-regex       BGP AS path regular expression for entries to match
best              Show longest matching route
brief             Display brief output
+ community        Identifier for community (can include wildcards)
community-name     Name of configured community policy to match
damping           Show entries subjected to particular kind of route
damping
detail            Display detailed output
exact             Show routes that match exactly
extensive          Display extensive output
hidden            Show hidden entries
inactive          Show inactive entries
label             Label of entry in MPLS routing table
label-switched-path Name of LSP tunnel associated with entries
next-hop           IP address of next hop that is destination for entries
no-community       Show entries with no associated community
output            Show entries sent out a particular interface
range             Show all entries in prefix range
receive-protocol   Show information in format received from particular
routing protocol
source-gateway     IP address of source router for entries
table             Name of routing table
terse             Display terse output
|                Pipe through a command
```

Showing BGP Routes

You can combine the **show route protocol bgp** command with options such as **extensive** or **detail** to get more information; use the **hidden** switch to display prefixes that are hidden due to a lack of next-hop reachability or route filtering policy actions. You can filter the output based on community or AS-path regular expressions or based upon advertising gateway to quickly locate the route that concerns you.

Note that in JUNOS software, BGP routes are placed in the main routing table, which is called `inet.0`.

Viewing BGP Route Details

- Use the **detail** or **extensive** switches to display detailed information for matching routes

```
lab@London> show route protocol bgp 192.168.32.1 detail

inet.0: 11 destinations, 13 routes (11 active, 0 holddown, 0 hidden)
192.168.32.1/32 (1 entry, 1 announced)
  *BGP      Preference: 170/-101
            Next-hop reference count: 4
            Source: 10.222.3.2
            Next hop: 10.222.3.2 via fe-0/0/1.0, selected
            State: <Active Ext>
            Local AS: 64512 Peer AS: 65222
            Age: 7:35:05
            Task: BGP_65222.10.222.3.2+2549
            Announcement bits (3): 0-KRT 3-BGP.0.0.0.0+179 4-
Resolve tree 1
  AS path: 65222 I
  Localpref: 100
  Router ID: 192.168.32.1
```



Viewing Specific Routes

To see all information associated with a given prefix (for example, BGP information such as the AS path, origin, local preference, MED attributes, and community strings), use the command as shown on the slide. You also can use the **show route extensive** command to determine the reason why the prefix is hidden (for example, next hop unusable).

Review Questions

1. Describe the general purpose of routing policy.
2. How do you confirm the results of an import policy?
3. Describe the purpose and role of an IGP.
4. How can you confirm OSPF adjacency status?
5. How can you display only those routes that are learned by a certain protocol?
6. How do you display the BGP routes that are sent or received from a given peer?



This Chapter Discussed:

- Routing tables and route preferences;
- JUNOS software routing policy and monitoring its operation;
- Static routing;
- IGP operation and purpose;
- RIP configuration and operation;
- OSPF configuration and operation; and
- BGP overview and basic configuration.

Lab 5: Static and BGP Routing

- Configure and monitor static and basic BGP routing.



Lab 5: Static and BGP Routing

The slide shows the objective for this lab.



Operating Juniper Networks Routers in the Enterprise

Chapter 7: Adaptive Services

Chapter Objectives

- After successfully completing this chapter, you will be able to:
 - Describe the services architecture of J-series platforms
 - Compare packet filtering and stateful firewalls
 - Use J-Web to configure and monitor stateful firewalls
 - Describe NAT and PAT
 - Use J-Web to configure and monitor NAT and PAT
 - Describe IPSec tunnels and explain how they can provide a VPN service
 - Use J-Web to configure and monitor IPSec tunnels
 - Describe the purpose of an intrusion detection system
 - Describe the purpose of flow monitoring and accounting
 - List the primary features of J-series CoS support



This Chapter Discusses:

- The J-series services features and architecture;
- Packet filters and stateful firewalls;
- Network Address Translation (NAT) and Port Address Translation (PAT);
- IPSec VPN tunnels;
- Typical intrusion detection system (IDS) and flow monitoring applications; and
- J-series class-of-service (CoS) overview.

Agenda: Adaptive Services

- Overview of Adaptive Services Features and Architecture
- Configuration and Monitoring of Packet Filters
- Configuration and Monitoring of Stateful Firewalls
- Configuration and Monitoring of NAT and PAT
- Configuration and Monitoring of IPSec Tunnels
- Overview of IDS Capabilities
- Overview of Flow Monitoring and Accounting
- Overview of J-series CoS Support



Overview of Adaptive Services Features and Architecture

The slide lists the topics we cover in this chapter. We discuss the highlighted topic first.

Our Objectives

- Services can be complicated
 - Full coverage of stateful firewall, NAT, IPSec, IDS, and J-Flow is outside the scope of this class
 - Our goal is to provide basic services configuration and monitoring examples
 - Students should attend selected advanced classes for detailed coverage of JUNOS software services



Objectives

This chapter attempts to provide a foundation for basic services configuration, but advanced services configuration is beyond the scope of this class. Detailed coverage of JUNOS software services is available in advanced courses. See <http://www.juniper.net/training/> for a current list of courses.

Overview of Adaptive Services

- Several advanced packet-processing services are available
 - Stateful firewall
 - Network address/port translation
 - IPSec VPN tunnel
 - Intrusion detection
 - Flow monitor
 - Tunnel service



Adaptive Services Overview

The slide lists some of the advanced packet-processing services available on J-series platforms or on M-series platforms with an AS PIC or Adaptive Services Module. These advanced services are in addition to the packet-filtering and CoS features available on all platforms, which are also covered in this chapter. These advanced services include the following:

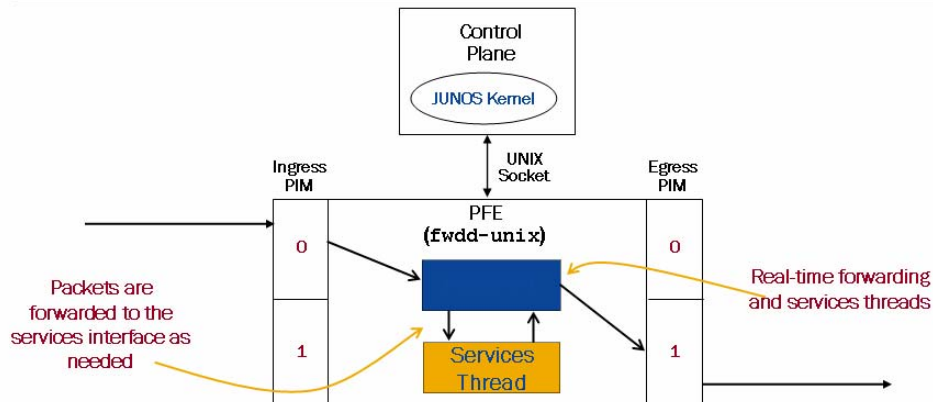
- *Stateful firewall:* This service stores state information about each IP flow and provides packet inspection based on that state and user configuration. A stateful firewall provides a higher level of security than a packet filter because it allows conditional acceptance of a packet based on compliance with protocol state.
- *NAT and PAT:* This service statically or dynamically translates IP addresses and protocol port numbers in a packet's header. NAT and PAT are typically used to translate between private and public IP addresses at the administrative borders of a network.
- *IPSec VPN tunnel:* Provides a secure virtual link between two gateways by encrypting an entire IP packet and placing the resulting encrypted data within the payload of a new IP packet. The resulting packet is then transmitted to the remote gateway where the outer packet is discarded, the payload is decrypted, and the original packet is forwarded.

Continued on next page.

Adaptive Services Overview (contd.)

- *Intrusion detection*: This service monitors per-flow state tables for protocol anomalies. Those anomalies are reported as possible intrusion attempts.
- *Flow monitor*: This service gathers flow-based statistics by statefully tracking IP flows and exporting standards-based v5 and v8 cflowd records.
- *Tunnel service*: This service encapsulates packets using several different Layer 2 and Layer 3 encapsulation mechanisms including the Multilink Point-to-Point Protocol (MLPPP), generic routing encapsulation (GRE), IP over IP, and Protocol Independent Multicast sparse mode (PIM SM).

J-series Services Architecture



- Services are provided by a software instantiation of the M-series and T-series Adaptive Services PIC
 - Manifested as a virtual service interface named `sp-0/0/0`
 - Handled as a real-time thread within the forwarding process

Virtual AS PIC

The software-based Packet Forwarding Engine (PFE) on J-series routers includes a software instantiation of the Adaptive Services PIC available for M-series and T-series routers. This virtual AS PIC is implemented as a real-time thread within the J-series forwarding process. Packets that require additional services processing are forwarded to the services thread for processing, while packets that do not require additional processing avoid the services thread. The services thread presents itself as a virtual `sp-0/0/0` interface in the JUNOS software.

Agenda: Adaptive Services

- Overview of Adaptive Services Features and Architecture
- ➔ **Configuration and Monitoring of Packet Filters**
- Configuration and Monitoring of Stateful Firewalls
- Configuration and Monitoring of NAT and PAT
- Configuration and Monitoring of IPSec Tunnels
- Overview of IDS Capabilities
- Overview of Flow Monitoring and Accounting
- Overview of J-series CoS Support



Configuration and Monitoring of Packet Filters

The slide highlights the topic we discuss next.

What Is Packet Filtering?

```

▶ Frame 3 (55 bytes on wire, 55 bytes captured)
▶ Ethernet II, Src: 00:0d:60:8b:6f:7f, Dst: 00:05:85:c7:53:d0
▶ Internet Protocol, Src Addr: 10.0.1.100 (10.0.1.100), Dst Addr: 10.0.1.70 (10.0.1.70)
▶ Transmission Control Protocol, Src Port: 1307 (1307), Dst Port: telnet (23), Seq: 0, Ack: 0,
▶ Telnet

```

Packet decode showing typical protocol fields

- Packet filters match packets based on:
 - Selected source or destination address
 - Protocol identifier, such as TCP, UDP, ICMP, or OSPF
 - Most other fields in the IP and transport protocol headers
- Filters can protect the router or downstream devices by blocking many types of DoS attacks
- Packet filters do not keep track of protocol state
 - Each packet is processed independently of previous or subsequent packets in a given flow
- Large-scale filter management GUI



Packet Filter Matches

The JUNOS software allows packet filters to be configured at the `[edit firewall filter filter-name]` CLI hierarchy. A packet filter is composed of one or more match/action pairs. Matches can be made on source or destination IP address, transport protocol (TCP or UDP) port numbers, and most other fields in the packet header. Possible actions include silently discarding the packet, rejecting the packet with an Internet Control Message Protocol (ICMP) notification, accepting the packet, counting the packet, sampling the packet, and logging the packet.

Applying Packet Filters

You apply packet filters to traffic transiting a particular router interface at the `[edit interfaces interface-name unit unit-number family inet filter]` CLI hierarchy. You can apply filters in both the input and output directions. Filters applied to the `lo0` interface have a special meaning. They apply to all traffic destined for the router itself, regardless of on which physical interface they were received or for which router IP address they are destined. Filters applied to the `lo0` interface do not affect packets transiting the router, which greatly simplifies the task of securing the router itself.

Continued on next page.

Packet Filters Are Stateless

Packet filters do not maintain state. Instead, they process each packet independently of previous or subsequent packets in a given IP flow. This prevents them from conditionally permitting or denying a packet based on other packets in the IP flow.

Large-Scale Filter Management GUI

See Appendix B for brief discussion of new large-scale firewall filter management GUI.

Sample Packet Filter Configuration

Define filter

```
[edit]
lab@London# show firewall
filter Protect-Web-Server {
  term permit-HTTP {
    from {
      destination-address {
        10.222.3.100/32;
      }
      protocol tcp;
      destination-port http;
    }
    then accept;
  }
  term deny-All {
    then {
      discard;
    }
  }
}
```

Apply filter

```
[edit]
lab@London# show interfaces fe-0/0/1
description "to Amsterdam fe-0/0/1";
unit 0 {
  family inet {
    filter {
      output Protect-Web-Server;
    }
    address 10.222.3.1/24;
  }
}
```

Packet Filter Example

The slide demonstrates a simple packet filter that helps to protect a Web server behind *London's* fe-0/0/1 interface and with IP address 10.222.3.100. The packet filter is defined at the [edit firewall filter filter-name] CLI hierarchy and permits only TCP traffic with a destination port number of 80 and a destination IP address of 10.222.3.100. The filter is then applied in the outbound direction on the fe-0/0/1 interface.

Agenda: Adaptive Services

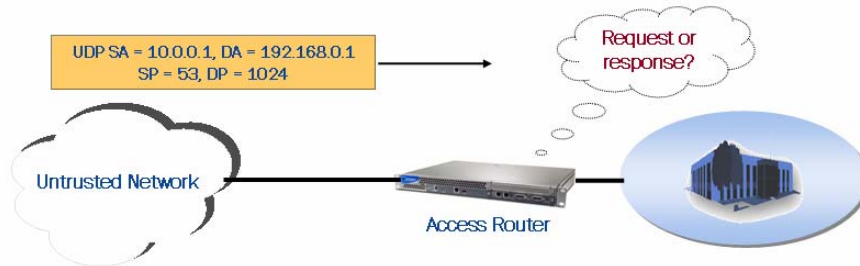
- Overview of Adaptive Services Features and Architecture
- Configuration and Monitoring of Packet Filters
- Configuration and Monitoring of Stateful Firewalls
- Configuration and Monitoring of NAT and PAT
- Configuration and Monitoring of IPSec Tunnels
- Overview of IDS Capabilities
- Overview of Flow Monitoring and Accounting
- Overview of J-series CoS Support



Configuration and Monitoring of Stateful Firewalls

The slide highlights the topic we discuss next.

Problems with Packet Filters



- Goal: Allowing incoming UDP from port 53 *only* when the traffic is in response to a previous DNS query
 - Problem: A basic packet filter will either allow or deny *all* incoming UDP from port 53
 - Solution: Use a stateful firewall

Lack of State

As mentioned earlier, packet filters do not maintain state. Instead, they process each packet independently of previous or subsequent packets in a given IP flow. This design prevents them from conditionally permitting or denying a packet based on other packets in the IP flow.

We only want to permit incoming UDP packets from port 53 if the traffic is a DNS response to a previous DNS query. A packet filter cannot provide this capability, but a stateful firewall can. A stateful firewall stores state information about each IP flow and provides packet inspection based on that state. A stateful firewall provides a higher level of security than a packet filter because it allows conditional acceptance of a packet based on compliance with protocol state.

In this example, a stateful firewall will only permit this packet if it has recently seen a UDP packet from 192.168.0.1 to 10.0.0.1 with a destination port of 53 and a source port of 1024.

Stateful Packet Inspection

Triggers new entry in the state table

Expect UDP SA = 10.0.0.1, DA = 192.168.0.1, SP = 53, DP = 1024 within n seconds

New protocol flow

UDP SA = 192.168.0.1, DA = 10.0.0.1, SP = 1024, DP = 53

Untrusted Network

Access Router

- Stores the protocol state by monitoring numerous fields in the protocol headers
 - New communications update the state table
 - Traffic that initiates communications allowed only when explicitly defined
 - Response traffic that relates to an established communication is permitted
- Provides higher level of security than packet filters

Juniper your Net

7-14

Copyright © 2007 Juniper Networks, Inc. Proprietary and Confidential www.juniper.net

Stateful Packet Inspection

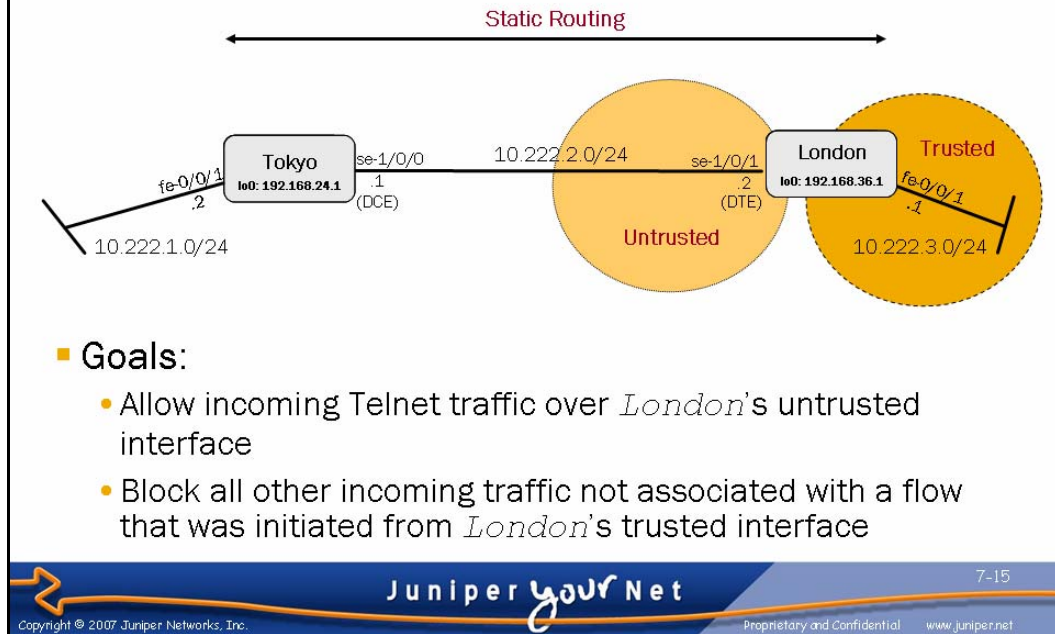
In this example, a stateful firewall is configured to permit outgoing DNS query traffic. It will only permit the corresponding DNS reply if it has an entry in its state table. The state table entry is created by the outgoing DNS query packet.

Higher Security

The ability to conditionally permit packets based on previous packets in the IP flow provides greater security than packet filters. In addition, stateful firewalls can monitor application-level data using an application-level gateway (ALG). An FTP ALG can monitor the data in an FTP control channel and use it to conditionally permit the corresponding FTP data channel.

Of course, the added security provided by stateful firewalls comes at a price. Additional memory and CPU resources are consumed to maintain and monitor state information. You can use packet filters and stateful firewalls together to provide higher security with minimal resources. Use packet filters to block some traffic and minimize the traffic that must be inspected by a stateful firewall.

Sample Topology: Stateful Firewall



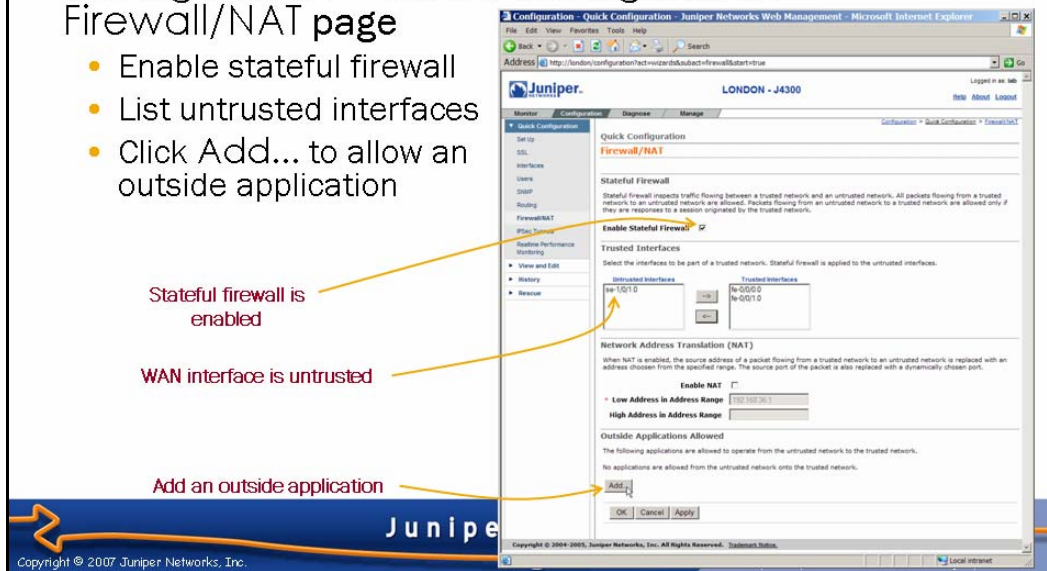
Stateful Firewall Example

We use the topology on the slide to demonstrate a stateful firewall. We will permit all traffic from *London's* trusted interface to its untrusted interface. Traffic from *London's* untrusted interface to its trusted interface will only be allowed if it is TCP port 23 (Telnet) or if it is a response to an existing IP flow.

Configuring Stateful Firewall: J-Web (1 of 2)

- Access the J-Web stateful firewall wizard at the Configuration > Quick Configuration > Firewall/NAT page

- Enable stateful firewall
- List untrusted interfaces
- Click Add... to allow an outside application



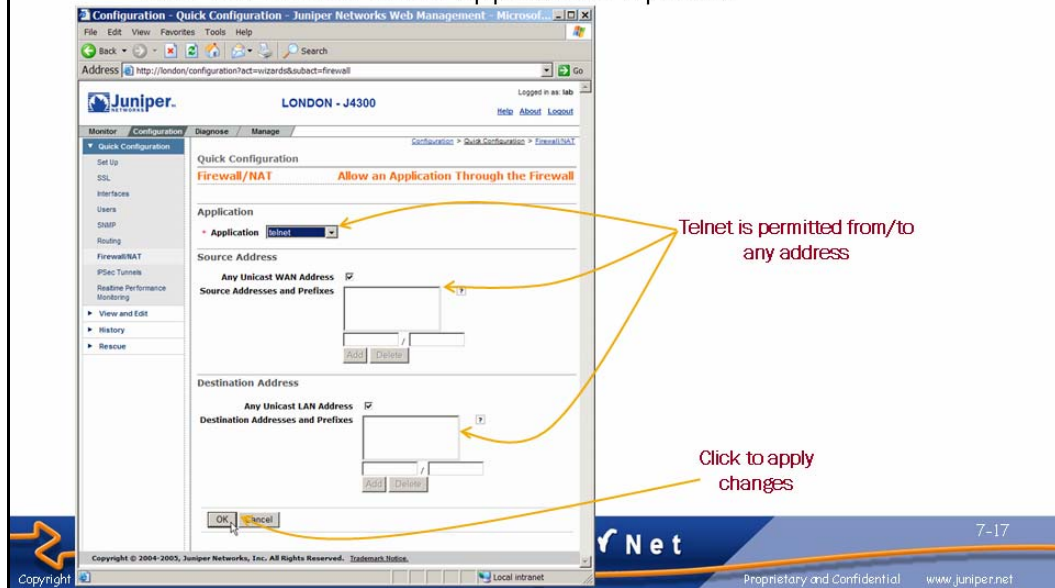
Configuring Stateful Firewall with J-Web

The J-Web stateful firewall wizard greatly simplifies basic stateful firewall configurations. Begin by navigating to the J-Web Configuration > Quick Configuration > Firewall/NAT page and checking the Enable Stateful Firewall checkbox. Select the untrusted interfaces, and then add any applications that should be allowed to initiate connections from the untrusted network to the trusted network without a corresponding state entry.

When you click the Add... button to add applications, the router applies the stateful firewall configuration before displaying the page that allows you to configure applications the router should allow from untrusted interfaces. Therefore, if you are accessing the router from an untrusted interface, you will lose connectivity to the router as soon as you click the Add... button.

Configuring Stateful Firewall: J-Web (2 of 2)

- Choose the desired application and click OK
 - Use the CLI for more application options



Allowing Outside Applications

You can now specify the application to be allowed along with the permitted source and destination addresses. The J-Web drop-down list provides a subset of supported applications. Use the CLI to access the complete list of supported applications. As of JUNOS software Release 8.1, the list includes the following:

- junos-bgp
- junos-biff
- junos-bootpc
- junos-bootps
- junos-citrix-winframe
- junos-citrix-winframe-udp
- junos-cvspserver
- junos-dce-rpc-portmap
- junos-dcerpc-endpoint-mapper-service
- junos-dcerpc-msexchange-directory-rfr
- junos-dcerpc-msexchange-directory-nsp
- junos-dcerpc-msexchange-information-store

Continued on next page.

Allowing Outside Applications (contd.)

- junos-dhcp-client
- junos-dhcp-server
- junos-dns-tcp
- junos-dns-udp
- junos-finger
- junos-ftp
- junos-h323
- junos-http
- junos-https
- junos-icmp-all
- junos-icmp-ping
- junos-ident
- junos-iiop-java
- junos-iiop-orbix
- junos-ike
- junos-imap
- junos-imaps
- junos-ipsec-esp
- junos-ldap
- junos-ldp-tcp
- junos-ldp-udp
- junos-netbios-datagram
- junos-netbios-name-tcp
- junos-netbios-name-udp
- junos-netbios-session
- junos-netshow
- junos-nfsd-tcp
- junos-nfsd-udp
- junos-nntp
- junos-ntalk
- junos-ntp
- junos-pop3
- junos-printer
- junos-radicat
- junos-radius
- junos-realaudio
- junos-rexec
- junos-rip
- junos-rlogin
- junos-rpc-portmap-tcp
- junos-rpc-portmap-udp
- junos-rpc-services-tcp
- junos-rpc-services-udp
- junos-rsh
- junos-rtsp
- junos-smtp
- junos-snmp-get
- junos-snmp-get-next
- junos-snmp-response
- junos-snmp-trap
- junos-snpp
- junos-sqlnet
- junos-ssh
- junos-syslog
- junos-tacacs
- junos-tacacs-ds
- junos-talk-tcp
- junos-talk-udp
- junos-telnet
- junos-tftp
- junos-traceroute
- junos-traceroute-ttl-1
- junos-who
- junos-xnm-clear-text
- junos-xnm-ssl

The Resulting Configuration (1 of 4)

- The CLI's **compare** function calls out changes made to the baseline configuration in support of stateful firewall

```
[edit]
lab@London# show | compare rollback 2
[edit interfaces]
+ sp-0/0/0 {
+   unit 0 {
+     family inet;
+   }
+ }
[edit interfaces se-1/0/1 unit 0 family inet]
+ service {
+   input {
+     service-set jweb-wan-sfw-service-set;
+   }
+   output {
+     service-set jweb-wan-sfw-service-set;
+   }
+ }
...

```

Service interface configuration

The service set is applied to the untrusted interface



The Results: Part 1

The CLI configuration created by the J-Web stateful firewall wizard is somewhat complicated. First, an `sp-0/0/0` interface is configured with a single logical unit and family `inet`. Next, the service set is applied to both the input and output directions of the untrusted interface.

The Resulting Configuration (2 of 4)

```

. . .
[edit]
+ services {
+   stateful-firewall {
+     rule jweb-sfw-to-wan {
+       match-direction output;
+       term jweb-apply-alg {
+         from {
+           application-sets junos-algs-outbound;
+         }
+         then {
+           accept;
+         }
+       }
+       term jweb-accept-all {
+         then {
+           accept;
+         }
+       }
+     }
+   }
+ }
. . .

```

Rule set for egress traffic

Dynamically created application gateway tracks dynamic ports used by some applications, for example, active FTP

All new outgoing traffic is accepted



The Results: Part 2

An egress stateful firewall rule allows all outgoing traffic and dynamically creates states for all known application protocols.

The Resulting Configuration (3 of 4)

```

+ . . .
+ rule jweb-sfw-from-wan {
+     match-direction input;
+     term jweb-wan-app-0 {
+         from {
+             source-address {
+                 any-unicast;
+             }
+             destination-address {
+                 any-unicast;
+             }
+             applications junos-telnet;
+         }
+         then {
+             accept;
+         }
+     }
+     term jweb-discard-all {
+         then {
+             discard;
+         }
+     }
+ }
+ . . .

```

Rule set for ingress traffic

Incoming Telnet traffic is allowed

All other state-initiating traffic is discarded!

Be careful when modifying configuration stanzas that J-Web created!



The Results: Part 3

The ingress stateful firewall rule allows Telnet traffic with any unicast source and destination IP addresses. Otherwise, traffic is denied. The ingress rule only applies to traffic that does not match an existing state entry.

Warning: While you can modify firewall rules and terms that a J-Web Quick Configuration wizard created, if you make modifications beyond the capabilities of the wizard, J-Web will overwrite your changes the next time somebody uses that wizard to modify the same section of the configuration. For example, the Firewall wizard can only specify one application per term—Telnet, in the example on the slide. If you add FTP and SSH to the *jweb-wan-app-0* term, these applications will vanish the next time somebody uses the Quick Configuration wizard on this rule.

The Resulting Configuration (4 of 4)

```
. . .
+   service-set jweb-wan-sfw-service-set {
+       stateful-firewall-rules jweb-sfw-to-wan;
+       stateful-firewall-rules jweb-sfw-from-wan;
+       interface-service {
+           service-interface sp-0/0/0;
+       }
+   }
```

Service set definition and
service interface linking

The Results: Part 4

A service set is defined to link the inbound and outbound rules to the virtual `sp-0/0/0` interface. This is the service set that was previously applied to the untrusted `se-1/0/1` interface.

Monitoring Stateful Firewall: J-Web (1 of 2)

- Monitor stateful firewall flows with J-Web at the Monitor > Firewall > Stateful Firewall page
 - Shows state associated with ingress Telnet session and an egress ping

Copyright © 2007 Juniper Networks, Inc.

Protocol	Source IP	Source Port	Destination IP	Destination Port	Flow State	Direction	Frames
ICMP	10.222.1.2	0	10.222.3.2		Watch	Inbound	39
ICMP	10.222.3.2	8	10.222.1.2		Watch	Outbound	39
TCP	10.222.2.1	2271	10.222.3.2	23	Forward	Inbound	27
TCP	10.222.3.2	23	10.222.2.1	2271	Forward	Outbound	21

Stateful Firewall Monitoring

You can view the current flows seen by a stateful firewall using the J-Web Monitor > Firewall > Stateful Firewall page. You can gather equivalent information at the CLI using the commands **show services stateful-firewall flows** and **show services stateful-firewall conversations**. The flow state will be one of the following:

- Drop*: Drops all packets in the flow without response;
- Forward*: Forwards the packet in the flow without inspecting it;
- Reject*: Drops all packets in the flow with response; or
- Watch*: Inspects packets in the flow.

Monitoring Stateful Firewall: J-Web (2 of 2)

- Expand entries for additional details
 - Requestor/responder role, timer values, byte counts

Connectionless protocol response timer

Response traffic permitted

Ping initiated by 10.222.3.2

Protocol	Source IP	Source Port	Destination IP	Destination Port	Flow State	Direction	Frames
ICMP	10.222.1.2	0	10.222.3.2		Watch	Inbound	39
		Byte Count	3276			TCP Window Size	
		Flow Role	Responder			Flow Timeout	30
ICMP	10.222.3.2	8	10.222.1.2		Watch	Outbound	39
		Byte Count	3276			TCP Window Size	
		Flow Role	Master			Flow Timeout	30
TCP	10.222.2.1	2271	10.222.3.2	23	Forward	Inbound	27
TCP	10.222.3.2	23	10.222.2.1	2271	Forward	Outbound	21

Detailed Flow Information

You can obtain detailed information on each IP flow by clicking the plus sign (+). The slide shows the detail for both the inbound and outbound directions of the ICMP echo flow. You can confirm that the ping was initiated from *London*'s trusted interface because it has the Master role. The corresponding response state has a Responder role.

Controlling Stateful Firewall: CLI

- Use the `clear services stateful-firewall` command to remove flows

```
lab@London> clear services stateful-firewall flows ?
```

Possible completions:

<[Enter]>	Execute this command
destination-port	Destination port to use as filter
destination-prefix	Destination prefix to use as filter
interface	Name of adaptive services interface
protocol	IP protocol type to use as filter
service-set	Name of service set
source-port	Source port to use as filter
source-prefix	Source prefix to use as filter
	Pipe through a command

Switches control which flows are cleared

```
lab@London> clear services stateful-firewall flows protocol icmp
```

```
Interface  Service set
sp-0/0/0   jweb-wan-sfw-service-set
```

Flow removed
3

Removing State Entries

You can use the `clear services stateful-firewall` CLI command to force the removal of flows from the state table. You can specify various switches to control which flows are cleared.

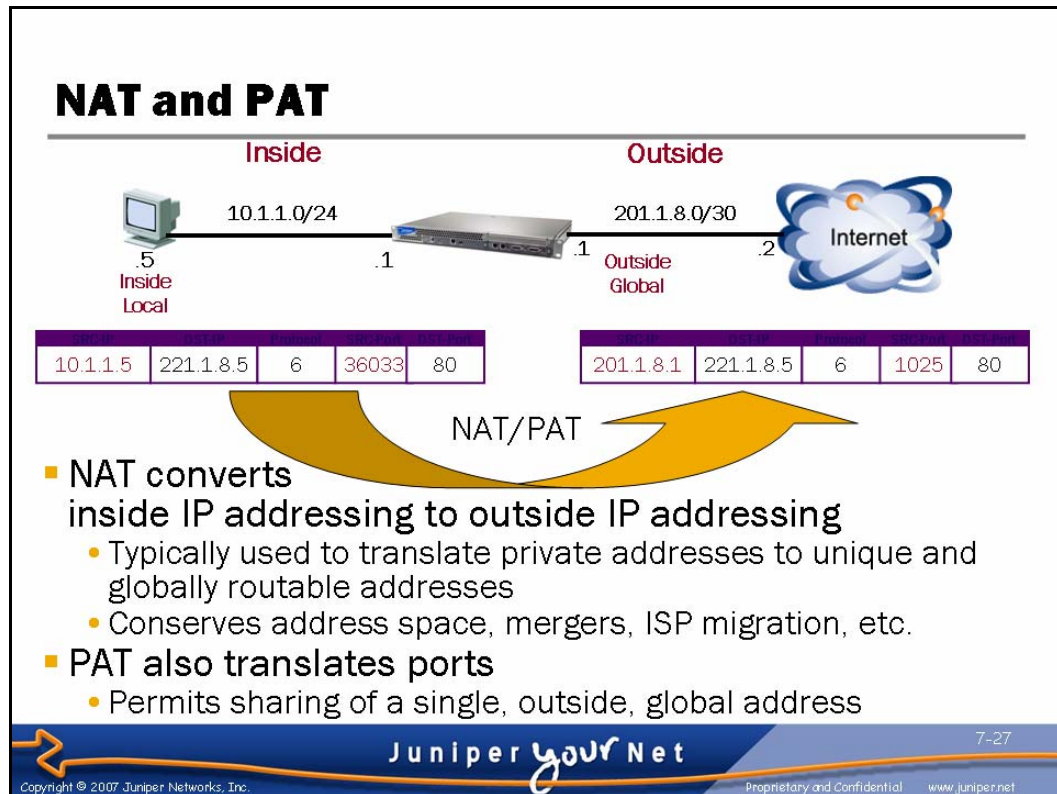
Agenda: Adaptive Services

- Overview of Adaptive Services Features and Architecture
- Configuration and Monitoring of Packet Filters
- Configuration and Monitoring of Stateful Firewalls
- ➔ **Configuration and Monitoring of NAT and PAT**
- Configuration and Monitoring of IPSec Tunnels
- Overview of IDS Capabilities
- Overview of Flow Monitoring and Accounting
- Overview of J-series CoS Support



Configuration and Monitoring of NAT/PAT

The slide highlights the topic we discuss next.



Network Address Translation

NAT is typically used to dynamically convert from private IP addresses to unique and globally routable public IP addresses. This process is done by rewriting the source IP address of packets traveling from the inside to outside networks. NAT helps to conserve IP address space by only using a global IP address for those hosts that must talk to the Internet. It is also very useful during transitions, such as a merger or ISP migration, when an organization must move from one block of IP addresses to another.

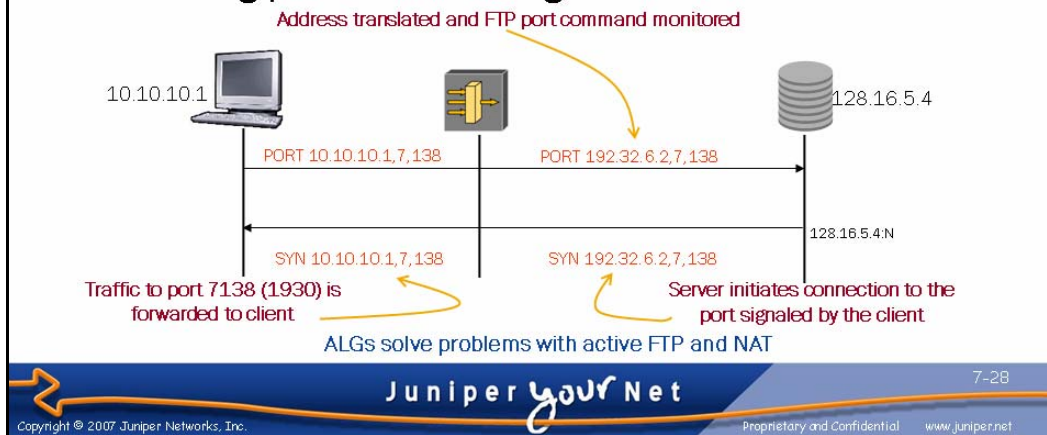
Port Address Translation

PAT modifies the source UDP or TCP ports of outgoing packets. When combined with NAT, PAT allows a large number of inside IP addresses to share a single outside IP address. Again, this helps to conserve globally unique IP addresses.

Both NAT and PAT must maintain a state table similar to a stateful firewall. This state table allows them to track which IP address and port numbers to use for a given flow.

Application-Level Gateways

- NAT *breaks* some applications because they signal dynamic ports within the application layer data
 - Examples include DNS, FTP, etc.
- ALGs dynamically create new translations by monitoring protocol exchanges



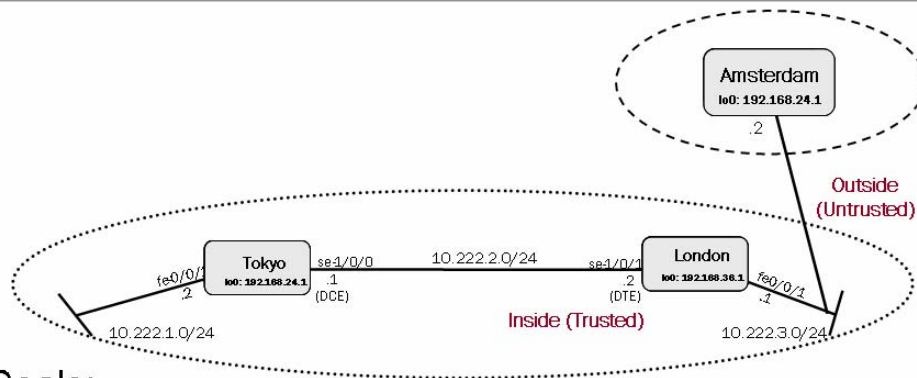
NAT Breaks Some Applications

Applications that imbed IP address or port information inside the application data portion of the packet cause problems for NAT. FTP, DNS, RealPlayer, and H.323 are just a few of the protocols that have this problem.

Application-Level Gateways to the Rescue

ALGs intercept and translate application layer signaling payloads to avoid these issues. The example on the slide shows an FTP ALG that monitors and translates the FTP data channel information that appears within an FTP port command on the FTP control channel.

Sample NAT/PAT Topology



Goals:

- Ensure that traffic originating on the 10.222.2.0/24 subnet is delivered to *Amsterdam* with a 10.222.3.1 source address
- Assume that multiple sources will be active at the same time
- Permit all ingress traffic on the untrusted interface

NAT/PAT Example

We use the topology on the slide to demonstrate NAT and PAT. The *London* router will translate all traffic sourced from the 10.222.2.0/24 subnet to a source address of 10.222.3.1 before sending it out its fe-0/0/1 interface towards *Amsterdam*. Because multiple sources might be active at the same time, PAT will be configured to allow all sources to be translated to the single 10.222.3.1 IP address. Finally, all ingress traffic on the untrusted interface will be permitted.

Configuring NAT: J-Web (1 of 2)

- Access the NAT wizard at the Configuration > Quick Configuration > Firewall/NAT page

The screenshot shows the Juniper J-Web configuration interface for the Firewall/NAT wizard. The interface is titled "LONDON - J4300" and includes a navigation menu on the left. The main content area is divided into several sections:

- Stateful Firewall:** Contains the "Enable Stateful Firewall" checkbox, which is checked. An annotation points to this checkbox with the text "NAT/PAT requires stateful firewall".
- Trusted Interfaces:** A section for selecting trusted interfaces.
- Untrusted Interfaces:** A section for selecting untrusted interfaces. An annotation points to this section with the text "Outside interface marked as untrusted".
- Network Address Translation (NAT):** Contains the "Enable NAT" checkbox, which is checked. Below it, the "Low Address in Address Range" field is set to "10.222.3.1". An annotation points to this field with the text "NAT enabled with a 10.222.3.1/32 address pool".
- Outside Applications Allowed:** A section for defining allowed applications.

The interface also includes a "Save" button at the bottom left and a "Local Internet" icon at the bottom right.

Configuring NAT with J-Web

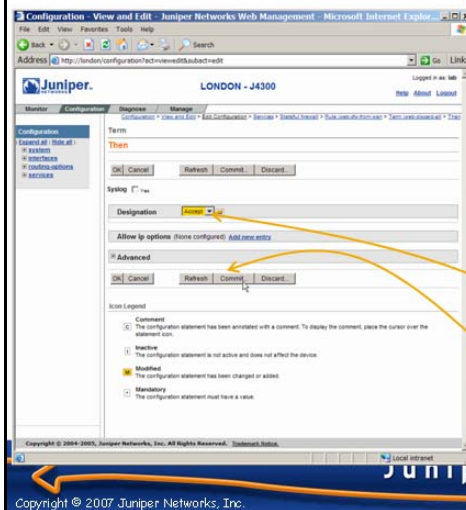
You can use the J-Web NAT wizard to easily implement the sample topology described on the previous slide. Access the NAT wizard by navigating to the Configuration > Quick Configuration > Firewall/NAT page. Beginning in JUNOS Release 7.6, you can configure NAT without configuring a stateful firewall. Whether or not a stateful firewall is configured, the interfaces listed as untrusted interfaces in the Firewall/NAT wizard are used as the outside interfaces for NAT. You can enable NAT and PAT by clicking the Enable NAT check box and defining the lower and upper bounds of the NAT IP address range. For PAT you must define only a single IP address in the Low Address in Address Range field.

Configuring NAT: J-Web (2 of 2)

- The related stateful firewall blocks all incoming traffic, by default

- Use the J-Web configuration editor or the CLI to modify so that all incoming traffic is accepted on the untrusted interface

- Select Accept at the Configuration > View and Edit > Edit Configuration > Services > Stateful firewall > Rule jweb-sfw-from-wan > Term jweb-discard-all > Then page



Value modified

Click to activate changes

Modifying the J-Web NAT Configuration

The configuration created by the wizard blocks all incoming traffic on the router's untrusted interface. We, however, wanted to permit all incoming traffic. We can do this by using the J-Web's View and Edit functionality or the CLI to modify the configuration created by the wizard. We simply change the result of the final stanza in the firewall rule to accept, rather than deny, traffic. We make this change at the J-Web Configuration > View and Edit > Edit Configuration > Services > Stateful firewall > Rule jweb-sfw-from-wan > Term jweb-discard-all > Then page. Do not forget to click Commit to activate your changes.

The Resulting Configuration (1 of 4)

- The CLI's `compare` function calls out changes made to the baseline configuration in support of NAT

```
[edit]
lab@London# show | compare rollback 2
[edit interfaces]
+   sp-0/0/0 {
+       unit 0 {
+           family inet;
+       }
+   }
[edit interfaces fe-0/0/1 unit 0 family inet]
+   service {
+       input {
+           service-set jweb-wan-sfw-service-set;
+       }
+       output {
+           service-set jweb-wan-sfw-service-set;
+       }
+   }
+   . . .
```

Service interface configuration

The service set is applied to the untrusted interface



The Results: Part 1

The CLI configuration created by the J-Web NAT wizard is simply an extension of the stateful firewall configuration we dissected earlier. First, an `sp-0/0/0` interface is configured with a single logical unit and family `inet`. Next, the service set is applied to both the input and output directions of the untrusted interface.

The Resulting Configuration (2 of 4)

```

...
[edit]
+ services {
+   stateful-firewall {
+     rule jweb-sfw-to-wan {
+       match-direction output;
+       term jweb-apply-alg {
+         from {
+           application-sets junos-algs-outbound;
+         }
+         then {
+           accept;
+         }
+       }
+       term jweb-accept-all {
+         then {
+           accept;
+         }
+       }
+     }
+   }
+ }
+ ...

```

Rule set for egress traffic

Dynamically created application gateway set tracks dynamic ports used by some applications—for example, active FTP

All other egress traffic is permitted



The Results: Part 2

An egress stateful firewall rule allows all outgoing traffic and dynamically creates states for all known application protocols.

The Resulting Configuration (3 of 4)

```

. . .
+       rule jweb-sfw-from-wan {
+         match-direction input;
+         term jweb-discard-all {
+           then {
+             accept;
+           }
+         }
+       }
+     nat {
+       pool jweb-nat-pool {
+         address-range low 10.222.3.1 high 10.222.3.1;
+         port automatic;
+       }
+       rule jweb-nat-to-wan {
+         match-direction output;
+         term jweb-nat-term {
+           then {
+             translated {
+               source-pool jweb-nat-pool;
+               translation-type source dynamic;
+             }
+           }
+         }
+       }
+     }
+   }
. . .

```

Rule set for ingress traffic

Modified to accept all ingress traffic; original term name unchanged

NAT pool allocates a single IP address

Output NAT rule set

All egress traffic without matching stateful firewall state is subjected to NAT in this example; no address match conditions specified

The Results: Part 3

The ingress stateful firewall rule was modified to allow all incoming traffic. Notice that the term name was not modified. This does not affect operation; term names are only for the user's reference. We now see that a NAT pool is created with a single IP address in the pool. A NAT rule is then defined to apply this pool. Because the NAT rule does not contain a *from* clause, translation will be applied to all egress traffic without matching firewall state.

The Resulting Configuration (4 of 4)

```

+ . . .
+   service-set jweb-wan-sfw-service-set {
+       stateful-firewall-rules jweb-sfw-to-wan;
+       stateful-firewall-rules jweb-sfw-from-wan;
+       nat-rules jweb-nat-to-wan;
+       interface-service {
+           service-interface sp-0/0/0;
+       }
+   }
+ }

```

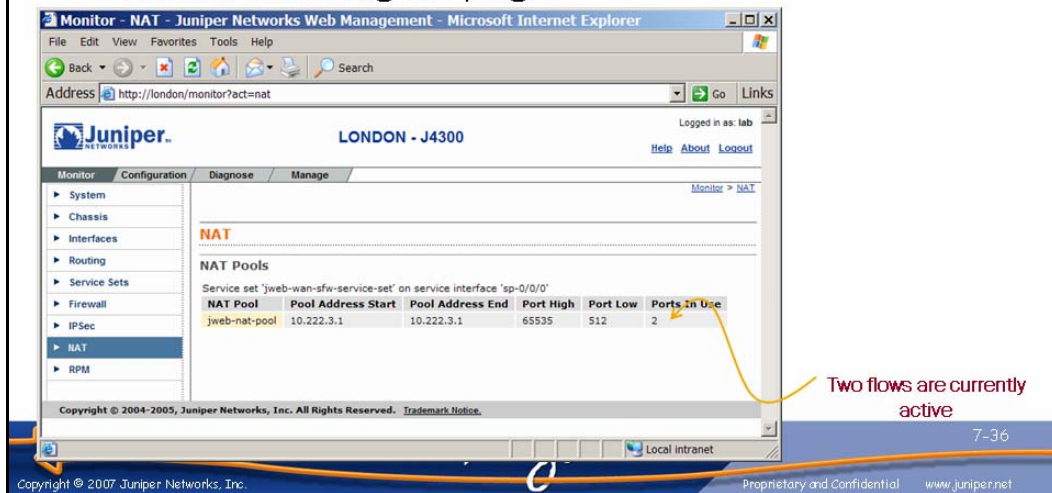
Service set definition and
service interface linking

The Results: Part 4

Just like the stateful firewall configuration, a service set is defined to link the inbound and outbound rules to the virtual `sp-0/0/0` interface. This is the service set that was previously applied to the untrusted `fe-0/0/1` interface. Note that this service set includes the two stateful firewall rules and the NAT rule.

Monitoring NAT

- Use J-Web Monitor > NAT page or CLI **show services nat pool** command to view NAT usage
 - Capture shows state associated with an egress Telnet session and an egress ping



Monitoring NAT

You can use the J-Web Monitor > NAT page or the CLI **show services nat pool** command to view NAT usage. The slide shows that two flows are currently active, which is the result of an egress ping and an egress Telnet that were sourced from *Tokyo* and destined for *Amsterdam*.

The following capture demonstrates the equivalent CLI output:

```
lab@London> show services nat pool
Interface: sp-0/0/0, Service set: jweb-wan-sfw-service-set
NAT pool          Address                      Port          Ports in use
jweb-nat-pool     10.222.3.1-10.222.3.1      65535-512    2
```

Agenda: Adaptive Services

- Overview of Adaptive Services Features and Architecture
- Configuration and Monitoring of Packet Filters
- Configuration and Monitoring of Stateful Firewalls
- Configuration and Monitoring of NAT and PAT
- ➔ **Configuration and Monitoring of IPSec Tunnels**
- Overview of IDS Capabilities
- Overview of Flow Monitoring and Accounting
- Overview of J-series CoS Support

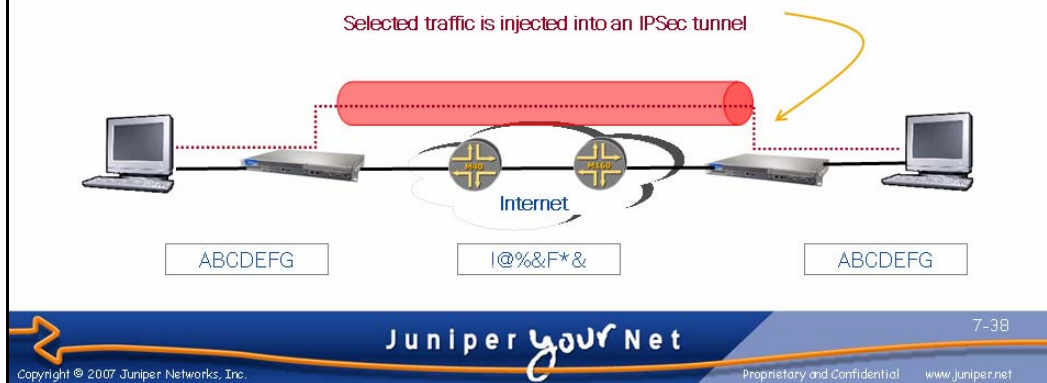


Configuration and Monitoring of IPSec Tunnels

The slide highlights the topic we discuss next.

IPSec VPN Tunnels

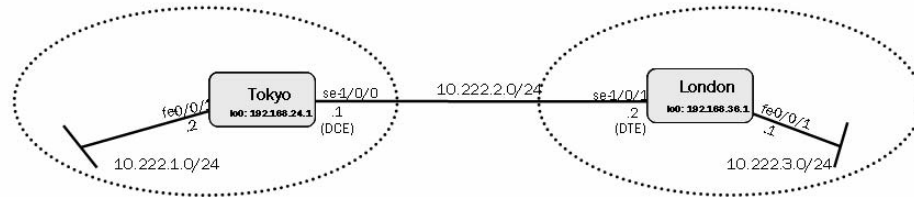
- Gateway-to-gateway tunnel
 - Traffic is protected only between IPSec gateways (router or firewall)
 - Protects transit traffic with encryption and authentication



Gateway-to-Gateway Tunnel

IPSec can provide its services between two security gateways. A security gateway can be a router or a firewall with IPSec capabilities. This method only protects the traffic between the gateways. The traffic between the hosts and the gateways is in cleartext. This method of IPSec tunneling essentially creates a secure virtual link between the two security gateways.

Sample IPsec Topology



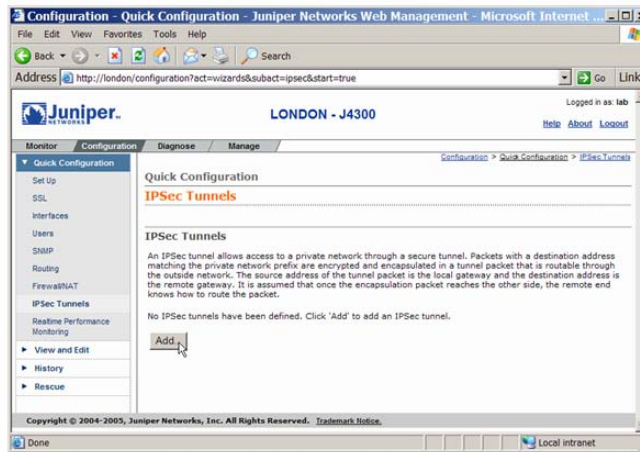
- Goal:
 - Ensure that traffic flowing between the LAN segments is placed into an IPsec tunnel for transport over the WAN

IPsec Example

We use the topology on the slide to demonstrate an IPsec tunnel. All traffic between *Tokyo's* and *London's* LAN segments will be encrypted before it is transmitted across the wide area connection.

Configuring IPSec Tunnels: J-Web (1 of 2)

- Access the IPSec wizard at the Configuration > Quick Configuration > IPSec Tunnels page
 - Click the Add... button to add an IPSec tunnel



J-Web IPSec Tunnel Configuration: Part 1

Begin configuring an IPSec tunnel using the J-Web IPSec wizard. Simply click the Add... button at the Configuration > Quick Configuration > IPSec Tunnels page.

Configuring IPSec Tunnels: J-Web (2 of 2)

- Define tunnel endpoints, the shared key, and the destination prefixes that should be tunneled

London's tunnel endpoints

Key value must match at both ends

Traffic to 10.222.1/24 is placed into the tunnel

J-Web IPSec Tunnel Configuration: Part 2

The tunnel endpoints are the local and remote IP addresses of the security gateway interfaces. In this case, that is the `se-1/0/1` IP address of *London* and the `se-1/0/0` IP address of *Tokyo*. The IKE Secret Key is a password value used to encrypt the data entering the tunnel. The remote tunnel endpoint must be configured with the same password to successfully decrypt the data received over the tunnel. Finally, the Private Prefix List describes the destinations that will be reached over the IPSec tunnel.

Monitoring IPSec Tunnels (1 of 2)

- To monitor IPSec tunnels:
 - Use the J-Web Monitor > IPSec page, or
 - Use the **show services ipsec-vpn** CLI command hierarchy
 - Click + to expand for more details
- IPSec tunnels are created on demand

Tunnel has not been used yet

Juniper Networks LONDON - J4300 Logged in as: admin
[Help](#) [About](#) [Logout](#)

Monitor Configuration Diagnose Manage **Alarms** Monitor > [IPSec](#)

IPSec

IPSec Tunnels

Service Set	Rule	Term	Local Gateway	Remote Gateway	Direction	Protocol
jweb-ipsec-tunnel-1	jweb-ipsec-rule-1	jweb-tunnel-all	10.222.2.2	10.222.2.1	No tunnels active	

IPSec Statistics
 No IPSec tunnels are configured or active.
 Click [here](#) to configure IPSec tunnels.

Juniper your Net 7-42
 Copyright © 2007 Juniper Networks, Inc. Proprietary and Confidential www.juniper.net

Monitoring IPSec with J-Web

The J-Web Monitor > IPSec page provides status and statistics of IPSec tunnels. The **show services ipsec-vpn** command hierarchy displays equivalent information using the CLI. (Note that the **show ipsec** and **show ike** command hierarchies only display information regarding the Encryption Services PIC and will not display information about IPSec VPNs on an AS PIC, Adaptive Services Module, or J-series platform.)

On-Demand Tunnels

By default, JUNOS software does not create IPSec tunnels until packets exist that must be routed over tunnels. This slide shows a configured tunnel not currently in use.

Monitoring IPSec Tunnels (2 of 2)

- An active IPSec tunnel:

IPSec Tunnels

Service Set	Rule	Term	Local Gateway	Remote Gateway	Direction	Protocol
jweb-ipsec-tunnel-1	jweb-ipsec-rule-1	jweb-tunnel-all	10.222.2.2	10.222.2.1	Inbound	ESP
jweb-ipsec-tunnel-1	jweb-ipsec-rule-1	jweb-tunnel-all	10.222.2.2	10.222.2.1	Outbound	ESP

IPSec Statistics

Service Set	Local Gateway	Remote Gateway	ESP Encrypted Bytes	ESP Decrypted Bytes	AH Input Bytes	AH Output Bytes
jweb-ipsec-tunnel-1	10.222.2.2	10.222.2.1	704	0	0	0

IKE Security

Remote Address	State	Initiator Cookie	Responder Cookie	Exchange Type
10.222.2.1	Matured	63e5fa3633065642	de93dc34c3c2999	Main

Active Tunnels

This slide depicts the monitoring status display of a tunnel once JUNOS software has activated the tunnel due to a traffic demand.

Agenda: Adaptive Services

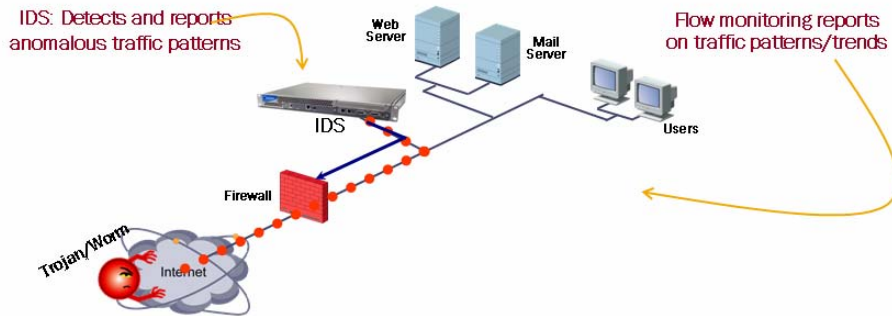
- Overview of Adaptive Services Features and Architecture
- Configuration and Monitoring of Packet Filters
- Configuration and Monitoring of Stateful Firewalls
- Configuration and Monitoring of NAT and PAT
- Configuration and Monitoring of IPSec Tunnels
- ➔ Overview of IDS Capabilities
- Overview of Flow Monitoring and Accounting
- Overview of J-series CoS Support



Overview of Intrusion Detection System Capabilities

The slide highlights the topic we discuss next.

Intrusion Detection System



- Stateful firewall also supports IDS functions based on anomaly detection
 - SYN floods, port scans, etc.

Intrusion Detection System

The J-series IDS analyzes the state table maintained for stateful firewall and NAT and PAT for unusual traffic patterns. These traffic patterns are often indicative of a network intrusion attempt. The J-series IDS functionality reports these anomalies for analysis and possible countermeasures.

Agenda: Adaptive Services

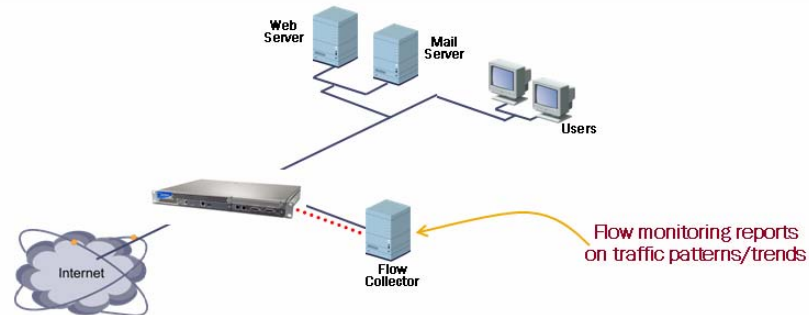
- Overview of Adaptive Services Features and Architecture
- Configuration and Monitoring of Packet Filters
- Configuration and Monitoring of Stateful Firewalls
- Configuration and Monitoring of NAT and PAT
- Configuration and Monitoring of IPSec Tunnels
- Overview of IDS Capabilities
- Overview of Flow Monitoring and Accounting
- Overview of J-series CoS Support



Overview of Flow Monitoring and Accounting

The slide highlights the topic we discuss next.

Flow Monitoring



- Flow monitoring statistically samples traffic and exports records to a collector
 - Baseline traffic patterns, capacity planning, anomaly detection, etc.

Flow Monitoring

J-series platforms can also sample traffic transiting the router and export standard cflowd records to a flow collector machine. You can then analyze these records to determine network traffic patterns. This analysis is often useful for capacity planning and anomaly detection.

Agenda: Adaptive Services

- Overview of Adaptive Services Features and Architecture
- Configuration and Monitoring of Packet Filters
- Configuration and Monitoring of Stateful Firewalls
- Configuration and Monitoring of NAT and PAT
- Configuration and Monitoring of IPSec Tunnels
- Overview of IDS Capabilities
- Overview of Flow Monitoring and Accounting
- Overview of J-series CoS Support



Overview of J-series CoS Support

The slide highlights the topic we discuss next

Overview of J-series CoS Features

- Key J-series CoS features include:
 - Behavior aggregate and multifield classification
 - Marker rewrite
 - DSCP, IP precedence, 802.1p
 - 8 transmission queues per logical interface
 - WRR scheduling with 5 priority levels (includes strict priority)
 - Weighted RED with 4 levels of drop priority
 - Rate limiting and shaping with burst capabilities
 - Frame Relay adaptive shaping based on BECN, DE-based classification and rewrite
 - Virtual channels to emulate Frame Relay VCs
- Full CoS coverage is outside the scope of this class

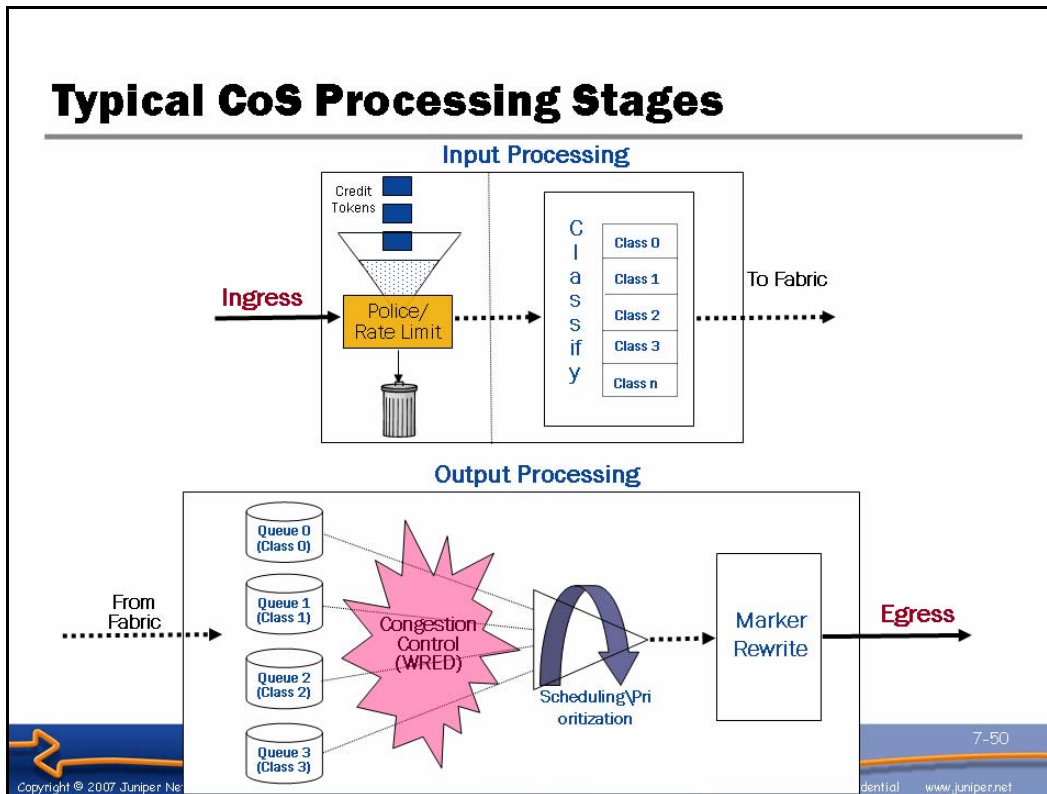


Key Class-of-Service Features

The slide lists the key CoS features on J-series platforms.

Beyond This Class

Full coverage of class-of-service concepts is beyond the scope of this class. See <http://www.juniper.net/training/> for a current list of courses.



Typical CoS Processing

This slide depicts typical CoS process stages on a modern router.

The discussion begins with ingress processing where received traffic is policed (rate limited). The policing function protects the network from abhorrent traffic patterns that might otherwise lead to congestion and possible disruption of service-level agreements (SLAs) associated with other users. In most cases, policing and rate-limiting actions are performed at the network's edge only.

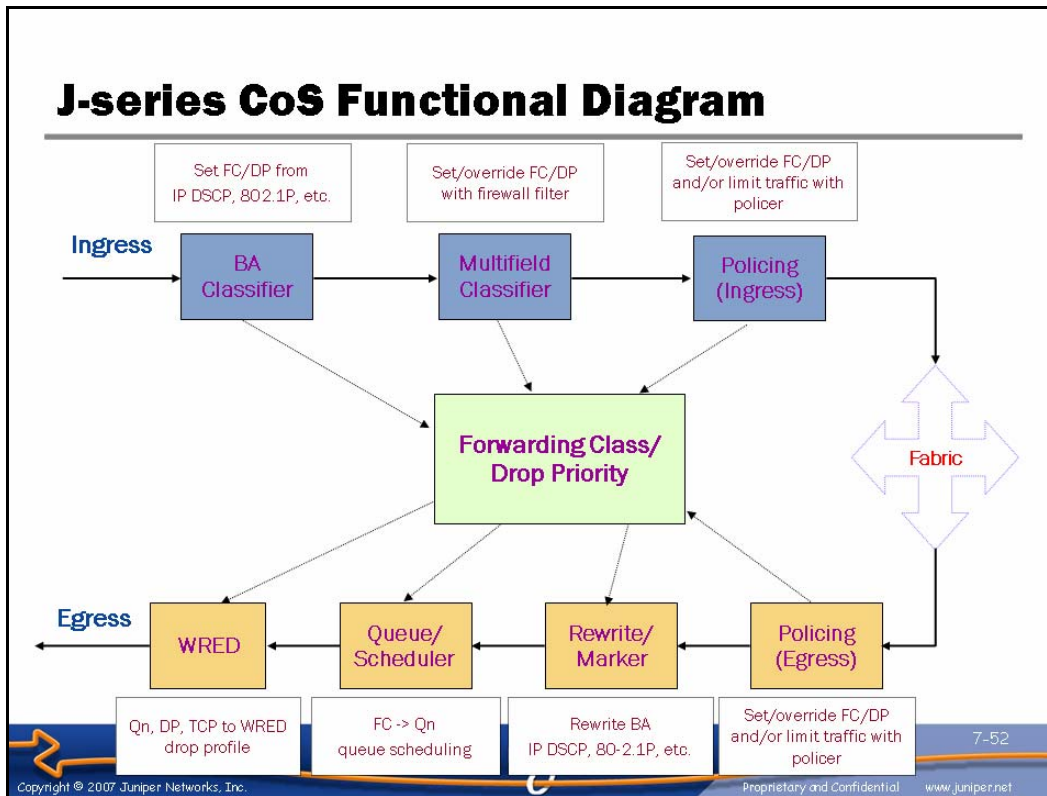
After policing comes traffic classification. Classification is a critical stage because being able to recognize different application streams is the foundation of being able to offer varying levels of service. Classification is necessary within all devices that handle the traffic because an end-to-end CoS design is contingent on the consistent handling of a given packet by all devices that interact with it. In most cases, customer-facing devices perform classification using some form of multifield classification. This type of classifier can inspect various fields within the packet to determine the nature of the traffic, albeit at the costs of increased computational burden. Once classified at the edge, various packet fields can be coded with a specific pattern to allow a computationally efficient behavior aggregate (BA) form of classification in downstream nodes.

Continued on next page.

Typical CoS Processing (contd.)

After transiting the switch fabric, a packet is normally placed into an outgoing queue, as identified during ingress classification. This queue is then subjected to some form of weighted round robin (WRR) servicing that factors in the bandwidth levels associated with each traffic class (or queue). Congestion avoidance is normally performed at this stage. Most often this takes the form of a random early detection (RED) algorithm that performs strategic discards to help prevent congestion.

The final stage of CoS processing involves the rewriting of specific markers, or fields, within the packet header to accommodate BA classification in downstream nodes. Some form of output shaping (not shown on the slide) might be used to reduce packet clumping and the resulting need for buffer space in downstream nodes.



J-series CoS Functional Block Diagram

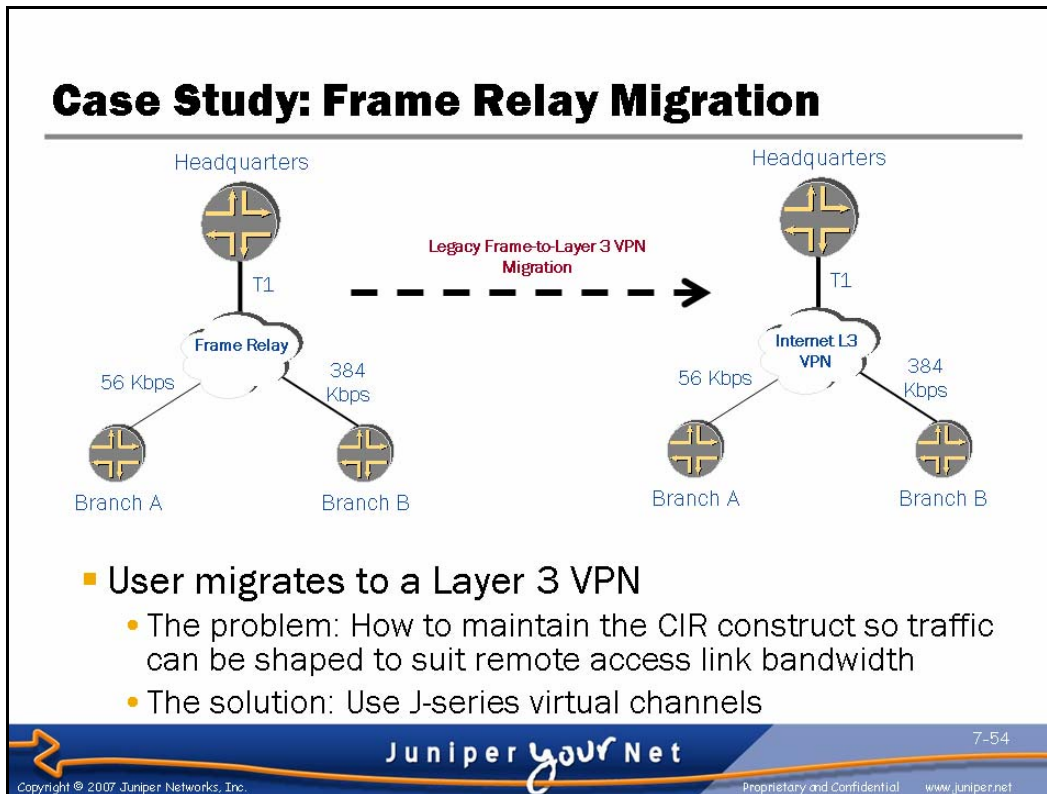
This slide provides a graphic depiction of the primary CoS processing stages on Juniper Networks J-series platforms. The graphic is broken down in the order of ingress traffic in the upper-left corner all the way to traffic egress at the lower-left corner. Note that based on the specific configuration, some CoS stages might be skipped, or the results of one stage might be overridden by a subsequent CoS processing action. In the following list, we describe each CoS processing stage:

- **Behavior aggregate (BA) classifier:** The first CoS processing stage occurs at ingress when traffic is classified according to a BA code point value in the form of IP precedence, DiffServ code points, or IEEE 802.1P priority values.
- **Multifield classifier:** This processing stage provides multifield classification capabilities based on a firewall filter. While it is unlikely that you will deploy both BA and multifield classification for the same traffic on the same chassis, it bears stressing that the results of multifield classification override the results of BA classification because of their processing order. The net result of traffic classification is the association of a forwarding class and loss-priority value for a particular packet based on the setting of various packet header fields.

Continued on next page.

J-series CoS Functional Block Diagram (contd.)

- *Policing*: Ingress policing limits the amount of traffic that can ingress the router, while egress policing shapes and limits the traffic volume that leaves the router. In most cases, ingress policing is only deployed on customer-facing edge routers. The ingress and egress policer stages are tied to the forwarding class/loss-priority block because policers can alter the packet's forwarding class or loss-priority settings when the policer's traffic profile is exceeded.
- *Marker rewrite*: This stage involves the ability to rewrite fields in the packet header to facilitate BA classification in intervening nodes. In most cases, edge routers perform multifield classification at ingress and then rewrite a given field in the packet so that core routers can efficiently classify traffic using a BA.
- *Queue scheduling*: Schedulers are used to service the queues associated with each forwarding class. Schedulers make use of WRR techniques to service each queue with the ability to support strict high, high, medium, and low priorities, in addition to configurable queue depths. The latter feature helps to limit maximum delays for time-sensitive traffic by favoring discard over deep queues.
- *WRED*: Congestion avoidance through a weighted random early detection (WRED) mechanism is also performed at this stage. J-series platforms support four levels of drop priority to weight discard based upon protocol type or drop priority, as assigned during ingress classification. The primary goal of congestion avoidance is to prevent global synchronization of TCP sessions, which is a condition where multiple sources begin retransmitting and backing off in unison, which in turn leads to oscillations of either too much or too little data.

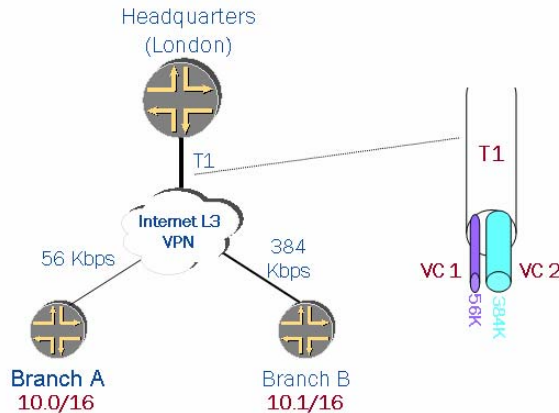


CoS Example

This case study examines how you can use J-series CoS functionality to avoid overloading low-speed branch office access lines when transiting traffic over a Layer 3 MPLS VPN.

Virtual Channels

```
[edit class-of-service]
lab@London# show
virtual-channels (
  branch_a;
  branch_b;
)
virtual-channel-groups (
  branch_ab (
    branch_a {
      scheduler-map sched;
      shaping-rate 56k;
    }
    branch_b {
      scheduler-map sched;
      shaping-rate 384k;
      default;
    }
  )
)
interfaces {
  se-1/0/1 {
    unit 0 {
      virtual-channel-group branch_ab;
    }
  }
}
```



VCs function like filter-based shapers

- Matching traffic is directed to a VC, each VC supports up to 8 queues and can be shaped to a maximum rate
- 64 VCs per logical interface, 512 per chassis

Virtual Channels as Filter-Based Shapers

This slide shows how J-series virtual channels (VCs) are used to emulate Frame Relay's ability to shape traffic on a per-DLCI (or VC) basis. By shaping VCs you can ensure that a central site with a high-capacity access link will not overwhelm the links of low-speed branch offices. Currently, VCs are only available on the J-series platform. The J-series platforms currently support a total of 4096 queues per chassis. The router allocates eight queues for each interface that has CoS enabled and eight queues per VC.

You begin by defining the each VC member at the [edit class-of-service virtual-channels] hierarchy. You then define a virtual-channel-group to link one or more VCs to an interface. As part of the group definition, you link each VC to a scheduler-map, which in turn links one or more scheduler definitions to provide per-queue WRR and traffic prioritization functions for up to eight queues. If wanted, you can also shape the aggregate rate of each VC using an absolute value or a percentage of interface bandwidth. Note that before migration to a provider-provisioned Layer 3 VPN, this type of shaping was achieved through the abstract of a Frame Relay DLCI. Note also that each VC group must have a default VC, which is used for traffic that is not directed to a specific VC. In this example traffic to Branch A is specifically directed to VC *branch_a*, while all other traffic is sent to the default VC. You must enable *per-unit-scheduling* at the device level of an interface that is to support VCs.

Continued on next page.

Virtual Channels as Filter-Based Shapers (contd.)

After defining the VCs and VC groups, you create a firewall filter to direct traffic to the correct VC using the virtual-channel action modifier, as shown. You can apply this filter to either the ingress or egress interface. The sample filter explicitly matches on destination prefixes associated with Branch A and directs them to the *branch_a* VC, while all remaining traffic is placed into the default VC. In this example the default VC is associated with Branch B:

```
[edit]
lab@London# show firewall
filter branch_ab_filter {
  term 1 {
    from {
      destination-address {
        10.0.0.0/16;
      }
    }
    then {
      virtual-channel branch_a;
      accept;
    }
  }
  term 2 {
    then accept;
  }
}
[edit]
lab@London# show interfaces se-1/0/1
per-unit-scheduler;
encapsulation cisco-hdlc;
unit 0 {
  family inet {
    filter {
      output branch_ab_filter;
    }
    address 10.222.2.2/24;
  }
}
```

Review Questions

1. How does a stateful firewall differ from a packet filter?
2. How do you configure PAT on an AS PIC or J-series platform? Describe a typical use for IPSec tunnels.
3. Describe a forwarding class, and list two ways that traffic can be classified.



This Chapter Discussed:

- The J-series services features and architecture;
- Packet filters and stateful firewalls;
- NAT and PAT;
- IPSec VPN tunnels;
- Typical IDS and flow monitoring applications; and
- J-series CoS overview.

Lab 6: Services

- Configure and monitor stateful firewall and NAT services.



Lab 6: Services

The slide shows the objective for this lab.



Operating Juniper Networks Routers—J-series

Appendix A: Supported PIMs

Supported PIMs

- The following tables show the latest PIM support:

Table 1—WAN PIM, Platform, and Software Compatibility

WAN PIMs	Description	Minimum JUNOS	
		J4300	J6300
JX-2T1-RJ48-S	2xT1 PIM	JUNOS 7.0*	JUNOS 7.0*
JX-2E1-RJ48-S	2xE1 PIM	JUNOS 7.0*	JUNOS 7.0*
JX-2Serial-S	2xSerial PIM	JUNOS 7.0*	JUNOS 7.0*
JX-1DS3-S	1xDS3 PIC	N/A	JUNOS 7.0
JX-1ADSL-A-S	1 Port ADSL Annex A PIM	JUNOS 7.2	JUNOS 7.2
JX-1ADSL-B-S	1 Port ADSL Annex B PIM	JUNOS 7.2	JUNOS 7.2
JX-1E3-S	1 Port E3 PIM	N/A	JUNOS 7.3
JX-4BRI-S-S	4xISDN BRI - S Interface	JUNOS 7.3	JUNOS 7.3
JX-4BRI-U-S	4xISDN BRI - U Interface	JUNOS 7.3	JUNOS 7.3
JX-2SHDSL-S	2-Port 2-wire G.SHDSL Interface	JUNOS 7.4	JUNOS 7.4

Table 2—Ethernet PIM, Platform, and Software Compatibility

Ethernet PIMs	Description	Minimum JUNOS	
		J4300	J6300
JX-2FE-TX-S	2xFE PIM	JUNOS 7.0*	JUNOS 7.0*



Supported PIMs

The tables on this slide show the Physical Interface Modules (PIMs) available for each J-series platform and the version of JUNOS software required to use them.



Operating Juniper Networks Routers—J-series

Appendix B: New Features

Chapter Objective

- After successfully completing this chapter, you will be aware of additional features in JUNOS software



Objective

This chapter highlights some recent features that might be of interest to students taking this class but that are not discussed in detail during the course. Consult Juniper Technical Documentation at <http://www.juniper.net/techpubs/> for more information about these features.

New Features (1 of 2)

- **Dialing features (ISDN):**
 - On-demand routing
 - Bandwidth on demand and ML-PPP
 - Dial backup
 - Dial-in
 - Dialer callback
- **J-Web GUI:**
 - DHCP monitoring and configuration
 - Firewall filter management
 - CoS quick configuration
- **Protocols:**
 - Full CLNS routing
 - cRTP



Dialing Infrastructure

JUNOS software now has flexible options for using ISDN as a backup connection.

J-Web GUI

- The Dynamic Host Configuration Protocol (DHCP) server functionality now has monitoring and quick-configuration wizard.
- Extensive firewall filter management GUI exists that is designed for thousands of filters and thousands of terms—it is intended to provide equivalent functionality to a well-known firewall vendor's router access list management product.
- CoS Quick Configuration wizard exists.

Protocols

- J-series routers can fully route Connectionless Network Service (CLNS) (although M-series T-series routers can perform IS-IS routing, they do not support CLNS). This capability enables providers to create CLNS VPNs that will scale much better than legacy point-to-point CLNS tunneling over IP core strategies.
- J-series routers support the Compressed Real-Time Protocol (cRTP) to improve efficiency of voice data over low-speed serial links.

New Features (2 of 2)

- Automatic scp (background file transfers)
- Transit packet capture



SCP Background Transfers

You can now configure secure copy (scp) in addition to FTP as the protocol for transferring bulk data from the router in an automated fashion. JUNOS software provides several options for managing and preloading SSH host-keys, to ensure that transfers do not fail because the destination host's key is not yet known.

Transit Packet Capture

For a long time, JUNOS software has had the capability to capture packets sourced from or destined to the router. On J-series routers, JUNOS software can now perform full packet capture on transit traffic—packets that it is forwarding from one host to another. In the past this was not possible because the packet capture functionality resides on the RE, which does not see transit packets. You can now configure J-series routers to mirror certain or all packets from an interface to the RE. Full packets are sent, including Layer 2 headers, and as much of the payload as you want. Captures are stored in libpcap (tcpdump) format, and you can manually transfer them from the router using the same methods you use to transfer any other files.