Cisco IOS Optimized Edge Routing Configuration Guide

Release 12.4

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

Cisco IOS Optimized Edge Routing Configuration Guide

About Cisco IOS and Cisco IOS XE Software Documentation

Last updated: August 6, 2008

This document describes the objectives, audience, conventions, and organization used in Cisco IOS and Cisco IOS XE software documentation, collectively referred to in this document as Cisco IOS documentation. Also included are resources for obtaining technical assistance, additional documentation, and other information from Cisco. This document is organized into the following sections:

- Documentation Objectives, page i
- Audience, page i
- Documentation Conventions, page ii
- Documentation Organization, page iii
- Additional Resources and Documentation Feedback, page xi

Documentation Objectives

Cisco IOS documentation describes the tasks and commands available to configure and maintain Cisco networking devices.

Audience

The Cisco IOS documentation set is i ntended for users who configure and maintain Cisco networking devices (such as routers and switches) but who may not be familiar with the configuration and maintenance tasks, the relationship among tasks, or the Cisco IOS commands necessary to perform particular tasks. The Cisco IOS documentation set is also intended for those users experienced with Cisco IOS who need to know about new features, new configuration options, and new software characteristics in the current Cisco IOS release.

Documentation Conventions

In Cisco IOS documentation, the term *router* may be used to refer to various Cisco products; for example, routers, access servers, and switches. These and other networking devices that support Cisco IOS software are shown interchangeably in examples and are used only for illustrative purposes. An example that shows one product does not necessarily mean that other products are not supported.

This section includes the following topics:

- Typographic Conventions, page ii
- Command Syntax Conventions, page ii
- Software Conventions, page iii
- Reader Alert Conventions, page iii

Typographic Conventions

Cisco IOS documentation uses the following typographic conventions:

Convention	Description
^ or Ctrl	Both the ^ symbol and Ctrl represent the Control (Ctrl) key on a keyboard. For example, the key combination **^D** or **Ctrl-D** means that you hold down the Control key while you press the D key. (Keys are indicated in capital letters but are not case sensitive.)
string	A string is a nonquoted set of characters shown in italics. For example, when setting a Simple Network Management Protocol (SNMP) community string to *public*, do not use quotation marks around the string; otherwise, the string will include the quotation marks.

Command Syntax Conventions

Cisco IOS documentation uses the following command syntax conventions:

Convention	Description
bold	Bold text indicates commands and keywords that you enter as shown.
italic	Italic text indicates arguments for which you supply values.
[x]	Square brackets enclose an optional keyword or argument.
|	A vertical line, called a pipe, indicates a choice within a set of keywords or arguments.
[x | y]	Square brackets enclosing keywords or arguments separated by a pipe indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments separated by a pipe indicate a required choice.
[x {y | z}]	Braces and a pipe within square brackets indicate a required choice within an optional element.

Software Conventions

Cisco IOS uses the following program code conventions:

Convention	Description
`Courier font`	Courier font is used for information that is displayed on a PC or terminal screen.
`Bold Courier font`	Bold Courier font indicates text that the user must enter.
< >	Angle brackets enclose text that is not displayed, such as a password. Angle brackets also are used in contexts in which the italic font style is not supported; for example, ASCII text.
!	An exclamation point at the beginning of a line indicates that the text that follows is a comment, not a line of code. An exclamation point is also displayed by Cisco IOS software for certain processes.
[ ]	Square brackets enclose default responses to system prompts.

Reader Alert Conventions

The Cisco IOS documentation set uses the following conventions for reader alerts:

Caution Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Note Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the manual.

Timesaver Means *the described action saves time*. You can save time by performing the action described in the paragraph.

Documentation Organization

This section describes the Cisco IOS documentation set, how it is organized, and how to access it on Cisco.com. Included are lists of configuration guides, command references, and supplementary references and resources that make up the documentation set. The following topics are included:

- Cisco IOS Documentation Set, page iv
- Cisco IOS Documentation on Cisco.com, page iv
- Configuration Guides, Command References, and Supplementary Resources, page v

Cisco IOS Documentation Set

Cisco IOS documentation consists of the following:

- Release notes and caveats provide information about platform, technology, and feature support for a release and describe severity 1 (catastrophic), severity 2 (severe), and severity 3 (moderate) defects in released Cisco IOS code. Review release notes before other documents to learn whether or not updates have been made to a feature.
- Sets of configuration guides and command references organized by technology and published for each standard Cisco IOS release.
  - Configuration guides—Compilations of documents that provide informational and task-oriented descriptions of Cisco IOS features.
  - Command references—Compilations of command pages that provide detailed information about the commands used in the Cisco IOS features and processes that make up the related configuration guides. For each technology, there is a single command reference that covers all Cisco IOS releases and that is updated at each standard release.
- Lists of all the commands in a specific release and all commands that are new, modified, removed, or replaced in the release.
- Command reference book for **debug** commands. Command pages are listed in alphabetical order.
- Reference book for system messages for all Cisco IOS releases.

Cisco IOS Documentation on Cisco.com

The following sections describe the documentation organization and how to access various document types.

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

New Features List

The New Features List for each release provides a list of all features in the release with hyperlinks to the feature guides in which they are documented.

Feature Guides

Cisco IOS features are documented in feature guides. Feature guides describe one feature or a group of related features that are supported on many different software releases and platforms. Your Cisco IOS software release or platform may not support all the features documented in a feature guide. See the Feature Information table at the end of the feature guide for information about which features in that guide are supported in your software release.

Configuration Guides

Configuration guides are provided by technology and release and comprise a set of individual feature guides relevant to the release and technology.

Command References

Command reference books describe Cisco IOS commands that are supported in many different software releases and on many different platforms. The books are provided by technology. For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or the *Cisco IOS Master Command List, All Releases*, at http://www.cisco.com/en/US/docs/ios/mcl/all_release/all_mcl.html.

Cisco IOS Supplementary Documents and Resources

Supplementary documents and resources are listed in Table 2 on page xi.

Configuration Guides, Command References, and Supplementary Resources

Table 1 lists, in alphabetical order, Cisco IOS and Cisco IOS XE software configuration guides and command references, including brief descriptions of the contents of the documents. The Cisco IOS command references are comprehensive, meaning that they include commands for both Cisco IOS software and Cisco IOS XE software, for all releases. The configuration guides and command references support many different software releases and platforms. Your Cisco IOS software release or platform may not support all these technologies.

For additional information about configuring and operating specific networking devices, go to the Product Support area of Cisco.com at http://www.cisco.com/web/psa/products/index.html.

Table 2 lists documents and resources that supplement the Cisco IOS software configuration guides and command references. These supplementary resources include release notes and caveats; master command lists; new, modified, removed, and replaced command lists; system messages; and the debug command reference.

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
Cisco IOS AppleTalk Configuration Guide *Cisco IOS XE AppleTalk Configuration Guide* *Cisco IOS AppleTalk Command Reference*	AppleTalk protocol.
Cisco IOS Asynchronous Transfer Mode Configuration Guide *Cisco IOS Asynchronous Transfer Mode Command Reference*	LAN ATM, multiprotocol over ATM (MPoA), and WAN ATM.

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
Cisco IOS Bridging and IBM Networking Configuration Guide *Cisco IOS Bridging Command Reference* *Cisco IOS IBM Networking Command Reference*	• Transparent and source-route transparent (SRT) bridging, source-route bridging (SRB), Token Ring Inter-Switch Link (TRISL), and token ring route switch module (TRRSM). • Data-link switching plus (DLSw+), serial tunnel (STUN), block serial tunnel (BSTUN); logical link control, type 2 (LLC2), synchronous data link control (SDLC); IBM Network Media Translation, including Synchronous Data Logical Link Control (SDLLC) and qualified LLC (QLLC); downstream physical unit (DSPU), Systems Network Architecture (SNA) service point, SNA frame relay access, advanced peer-to-peer networking (APPN), native client interface architecture (NCIA) client/server topologies, and IBM Channel Attach.
Cisco IOS Broadband and DSL Configuration Guide *Cisco IOS XE Broadband and DSL Configuration Guide* *Cisco IOS Broadband and DSL Command Reference*	Point-to-Point Protocol (PPP) over ATM (PPPoA) and PPP over Ethernet (PPPoE).
Cisco IOS Carrier Ethernet Configuration Guide *Cisco IOS Carrier Ethernet Command Reference*	Connectivity fault management (CFM), Ethernet Local Management Interface (ELMI), IEEE 802.3ad link bundling, Link Layer Discovery Protocol (LLDP), media endpoint discovery (MED), and operations, administration, and maintenance (OAM).
Cisco IOS Configuration Fundamentals Configuration Guide *Cisco IOS XE Configuration Fundamentals Configuration Guide* *Cisco IOS Configuration Fundamentals Command Reference*	Autoinstall, Setup, Cisco IOS command-line interface (CLI), Cisco IOS file system (IFS), Cisco IOS web browser user interface (UI), basic file transfer services, and file management.
Cisco IOS DECnet Configuration Guide *Cisco IOS XE DECnet Configuration Guide* *Cisco IOS DECnet Command Reference*	DECnet protocol.
Cisco IOS Dial Technologies Configuration Guide *Cisco IOS XE Dial Technologies Configuration Guide* *Cisco IOS Dial Technologies Command Reference*	Asynchronous communications, dial backup, dialer technology, dial-in terminal services and AppleTalk remote access (ARA), large scale dialout, dial-on-demand routing, dialout, modem and resource pooling, ISDN, multilink PPP (MLP), PPP, virtual private dialup network (VPDN).
Cisco IOS Flexible NetFlow Configuration Guide *Cisco IOS Flexible NetFlow Command Reference*	Flexible NetFlow.

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
Cisco IOS H.323 Configuration Guide	Gatekeeper enhancements for managed voice services, Gatekeeper Transaction Message Protocol, gateway codec order preservation and shutdown control, H.323 dual tone multifrequency relay, H.323 version 2 enhancements, Network Address Translation (NAT) support of H.323 v2 Registration, Admission, and Status (RAS) protocol, tokenless call authorization, and VoIP gateway trunk and carrier-based routing.
Cisco IOS High Availability Configuration Guide *Cisco IOS XE High Availability Configuration Guide* *Cisco IOS High Availability Command Reference*	A variety of High Availability (HA) features and technologies that are available for different network segments (from enterprise access to service provider core) to facilitate creation of end-to-end highly available networks. Cisco IOS HA features and technologies can be categorized in three key areas: system-level resiliency, network-level resiliency, and embedded management for resiliency.
Cisco IOS Integrated Session Border Controller Command Reference	A VoIP-enabled device that is deployed at the edge of networks. An SBC is a toolkit of functions, such as signaling interworking, network hiding, security, and quality of service (QoS).
Cisco IOS Intelligent Service Gateway Configuration Guide *Cisco IOS Intelligent Service Gateway Command Reference*	Subscriber identification, service and policy determination, session creation, session policy enforcement, session life-cycle management, accounting for access and service usage, session state monitoring.
Cisco IOS Interface and Hardware Component Configuration Guide *Cisco IOS XE Interface and Hardware Component Configuration Guide* *Cisco IOS Interface and Hardware Component Command Reference*	LAN interfaces, logical interfaces, serial interfaces, virtual interfaces, and interface configuration.
Cisco IOS IP Addressing Services Configuration Guide *Cisco IOS XE Addressing Services Configuration Guide* *Cisco IOS IP Addressing Services Command Reference*	Address Resolution Protocol (ARP), Network Address Translation (NAT), Domain Name System (DNS), Dynamic Host Configuration Protocol (DHCP), and Next Hop Address Resolution Protocol (NHRP).
Cisco IOS IP Application Services Configuration Guide *Cisco IOS XE IP Application Services Configuration Guide* *Cisco IOS IP Application Services Command Reference*	Enhanced Object Tracking (EOT), Gateway Load Balancing Protocol (GLBP), Hot Standby Router Protocol (HSRP), IP Services, Server Load Balancing (SLB), Stream Control Transmission Protocol (SCTP), TCP, Web Cache Communication Protocol (WCCP), User Datagram Protocol (UDP), and Virtual Router Redundancy Protocol (VRRP).
Cisco IOS IP Mobility Configuration Guide *Cisco IOS IP Mobility Command Reference*	Mobile ad hoc networks (MANet) and Cisco mobile networks.
Cisco IOS IP Multicast Configuration Guide *Cisco IOS XE IP Multicast Configuration Guide* *Cisco IOS IP Multicast Command Reference*	Protocol Independent Multicast (PIM) sparse mode (PIM-SM), bidirectional PIM (bidir-PIM), Source Specific Multicast (SSM), Multicast Source Discovery Protocol (MSDP), Internet Group Management Protocol (IGMP), and Multicast VPN (MVPN).

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
Cisco IOS IP Routing Protocols Configuration Guide *Cisco IOS XE IP Routing Protocols Configuration Guide* *Cisco IOS IP Routing Protocols Command Reference*	Border Gateway Protocol (BGP), multiprotocol BGP, multiprotocol BGP extensions for IP multicast, bidirectional forwarding detection (BFD), Enhanced Interior Gateway Routing Protocol (EIGRP), Interior Gateway Routing Protocol (IGRP), Intermediate System-to-Intermediate System (IS-IS), on-demand routing (ODR), Open Shortest Path First (OSPF), and Routing Information Protocol (RIP).
Cisco IOS IP SLAs Configuration Guide *Cisco IOS XE IP SLAs Configuration Guide* *Cisco IOS IP SLAs Command Reference*	Cisco IOS IP Service Level Agreements (IP SLAs).
Cisco IOS IP Switching Configuration Guide *Cisco IOS XE IP Switching Configuration Guide* *Cisco IOS IP Switching Command Reference*	Cisco Express Forwarding, fast switching, and Multicast Distributed Switching (MDS).
Cisco IOS IPv6 Configuration Guide *Cisco IOS XE IPv6 Configuration Guide* *Cisco IOS IPv6 Command Reference*	For IPv6 features, protocols, and technologies, go to the IPv6 "Start Here" document at the following URL: http://www.cisco.com/en/US/docs/ios/ipv6/configuration/guide/ip6-roadmap.html
Cisco IOS ISO CLNS Configuration Guide *Cisco IOS XE ISO CLNS Configuration Guide* *Cisco IOS ISO CLNS Command Reference*	ISO connectionless network service (CLNS).
Cisco IOS LAN Switching Configuration Guide *Cisco IOS XE LAN Switching Configuration Guide* *Cisco IOS LAN Switching Command Reference*	VLANs, Inter-Switch Link (ISL) encapsulation, IEEE 802.10 encapsulation, IEEE 802.1Q encapsulation, and multilayer switching (MLS).
Cisco IOS Mobile Wireless Gateway GPRS Support Node Configuration Guide *Cisco IOS Mobile Wireless Gateway GPRS Support Node Command Reference*	Cisco IOS Gateway GPRS Support Node (GGSN) in a 2.5-generation general packet radio service (GPRS) and 3-generation universal mobile telecommunication system (UMTS) network.
Cisco IOS Mobile Wireless Home Agent Configuration Guide *Cisco IOS Mobile Wireless Home Agent Command Reference*	Cisco Mobile Wireless Home Agent, an anchor point for mobile terminals for which mobile IP or proxy mobile IP services are provided.
Cisco IOS Mobile Wireless Packet Data Serving Node Configuration Guide *Cisco IOS Mobile Wireless Packet Data Serving Node Command Reference*	Cisco Packet Data Serving Node (PDSN), a wireless gateway that is between the mobile infrastructure and standard IP networks and that enables packet data services in a code division multiple access (CDMA) environment.
Cisco IOS Mobile Wireless Radio Access Networking Configuration Guide *Cisco IOS Mobile Wireless Radio Access Networking Command Reference*	Cisco IOS radio access network products.

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
Cisco IOS Multiprotocol Label Switching Configuration Guide *Cisco IOS XE Multiprotocol Label Switching Configuration Guide* *Cisco IOS Multiprotocol Label Switching Command Reference*	MPLS Label Distribution Protocol (LDP), MPLS Layer 2 VPNs, MPLS Layer 3 VPNs, MPLS Traffic Engineering (TE), and MPLS Embedded Management (EM) and MIBs.
Cisco IOS Multi-Topology Routing Configuration Guide *Cisco IOS Multi-Topology Routing Command Reference*	Unicast and multicast topology configurations, traffic classification, routing protocol support, and network management support.
Cisco IOS NetFlow Configuration Guide *Cisco IOS XE NetFlow Configuration Guide* *Cisco IOS NetFlow Command Reference*	Network traffic data analysis, aggregation caches, export features.
Cisco IOS Network Management Configuration Guide *Cisco IOS XE Network Management Configuration Guide* *Cisco IOS Network Management Command Reference*	Basic system management; system monitoring and logging; troubleshooting, logging, and fault management; Cisco Discovery Protocol; Cisco IOS Scripting with Tool Control Language (Tcl); Cisco networking services (CNS); DistributedDirector; Embedded Event Manager (EEM); Embedded Resource Manager (ERM); Embedded Syslog Manager (ESM); HTTP; Remote Monitoring (RMON); SNMP; and VPN Device Manager Client for Cisco IOS Software (XSM Configuration).
Cisco IOS Novell IPX Configuration Guide *Cisco IOS XE Novell IPX Configuration Guide* *Cisco IOS Novell IPX Command Reference*	Novell Internetwork Packet Exchange (IPX) protocol.
Cisco IOS Optimized Edge Routing Configuration Guide *Cisco IOS Optimized Edge Routing Command Reference*	Optimized edge routing (OER) monitoring, policy configuration, routing control, logging and reporting, and VPN IPsec/generic routing encapsulation (GRE) tunnel interface optimization.
Cisco IOS Quality of Service Solutions Configuration Guide *Cisco IOS XE Quality of Service Solutions Configuration Guide* *Cisco IOS Quality of Service Solutions Command Reference*	Class-based weighted fair queuing (CBWFQ), custom queuing, distributed traffic shaping (DTS), generic traffic shaping (GTS), IP- to-ATM class of service (CoS), low latency queuing (LLQ), modular QoS CLI (MQC), Network-Based Application Recognition (NBAR), priority queuing, Security Device Manager (SDM), Multilink PPP (MLPPP) for QoS, header compression, AutoQoS, QoS features for voice, Resource Reservation Protocol (RSVP), weighted fair queuing (WFQ), and weighted random early detection (WRED).
Cisco IOS Security Configuration Guide *Cisco IOS XE Security Configuration Guide* *Cisco IOS Security Command Reference*	Access control lists (ACLs), authentication, authorization, and accounting (AAA), firewalls, IP security and encryption, neighbor router authentication, network access security, network data encryption with router authentication, public key infrastructure (PKI), RADIUS, TACACS+, terminal access security, and traffic filters.

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
Cisco IOS Service Selection Gateway Configuration Guide *Cisco IOS Service Selection Gateway Command Reference*	Subscriber authentication, service access, and accounting.
Cisco IOS Software Activation Configuration Guide *Cisco IOS Software Activation Command Reference*	An orchestrated collection of processes and components to activate Cisco IOS software feature sets by obtaining and validating Cisco software licenses.
Cisco IOS Software Modularity Installation and Configuration Guide *Cisco IOS Software Modularity Command Reference*	Installation and basic configuration of software modularity images, including installations on single and dual route processors, installation rollbacks, software modularity binding, software modularity processes and patches.
Cisco IOS Terminal Services Configuration Guide *Cisco IOS Terminal Services Command Reference* *Cisco IOS XE Terminal Services Command Reference*	DEC, local-area transport (LAT), and X.25 packet assembler/disassembler (PAD).
Cisco IOS Virtual Switch Command Reference	Virtual switch redundancy, high availability, and packet handling; converting between standalone and virtual switch modes; virtual switch link (VSL); Virtual Switch Link Protocol (VSLP). **Note** For information about virtual switch configuration, refer to the product-specific software configuration information for the Cisco Catalyst 6500 series switch or for the Metro Ethernet 6500 series switch.
Cisco IOS Voice Configuration Library *Cisco IOS Voice Command Reference*	Cisco IOS support for voice call control protocols, interoperability, physical and virtual interface management, and troubleshooting. The library includes documentation for IP telephony applications.
Cisco IOS VPDN Configuration Guide *Cisco IOS XE VPDN Configuration Guide* *Cisco IOS VPDN Command Reference*	Layer 2 Tunneling Protocol (L2TP) dial-out load balancing and redundancy, L2TP extended failover, L2TP security VPDN, multihop by Dialed Number Identification Service (DNIS), timer and retry enhancements for L2TP and Layer 2 Forwarding (L2F), RADIUS Attribute 82: tunnel assignment ID, shell-based authentication of VPDN users, tunnel authentication via RADIUS on tunnel terminator.
Cisco IOS Wide-Area Networking Configuration Guide *Cisco IOS XE Wide-Area Networking Configuration Guide* *Cisco IOS Wide-Area Networking Command Reference*	Frame Relay, Layer 2 Tunneling Protocol Version 3 (L2TPv3), Link Access Procedure, Balanced (LAPB), Switched Multimegabit Data Service (SMDS), and X.25.
Cisco IOS Wireless LAN Configuration Guide *Cisco IOS Wireless LAN Command Reference*	Broadcast key rotation, IEEE 802.11x support, IEEE 802.1x authenticator, IEEE 802.1x local authentication service for Extensible Authentication Protocol-Flexible Authentication via Secure Tunneling (EAP-FAST), Multiple Basic Service Set ID (BSSID), Wi-Fi Multimedia (WMM) required elements, and Wi-Fi Protected Access (WPA).

Table 2 Cisco IOS Supplementary Documents and Resources

Document Title	Description
Cisco IOS Master Command List, All Releases	Alphabetical list of all the commands documented in all Cisco IOS releases.
Cisco IOS New, Modified, Removed, and Replaced Commands	List of all the new, modified, removed, and replaced commands for a Cisco IOS release.
Cisco IOS Software System Messages	List of Cisco IOS system messages and descriptions. System messages may indicate problems with your system; be informational only; or may help diagnose problems with communications lines, internal hardware, or the system software.
Cisco IOS Debug Command Reference	Alphabetical list of **debug** commands including brief descriptions of use, command syntax, and usage guidelines.
Release Notes and Caveats	Information about new and changed features, system requirements, and other useful information about specific software releases; information about defects in specific Cisco IOS software releases.
MIBs	Files used for network monitoring. To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator at the following URL: http://www.cisco.com/go/mibs
RFCs	Standards documents maintained by the Internet Engineering Task Force (IETF) that Cisco IOS documentation references where applicable. The full text of referenced RFCs may be obtained at the following URL: http://www.rfc-editor.org/

Additional Resources and Documentation Feedback

What's New in Cisco Product Documentation is published monthly and describes all new and revised Cisco technical documentation. The *What's New in Cisco Product Documentation* publication also provides information about obtaining the following resources:

- Technical documentation
- Cisco product security overview
- Product alerts and field notices
- Technical assistance

Cisco IOS technical documentation includes embedded feedback forms where you can rate documents and provide suggestions for improvement. Your feedback helps us improve our documentation.

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

Using the Command-Line Interface in Cisco IOS and Cisco IOS XE Software

Last updated: August 6, 2008

This document provides basic information about the command-line interface (CLI) in Cisco IOS and Cisco IOS XE software and how you can use some of the CLI features. This document contains the following sections:

- Initially Configuring a Device, page i
- Using the CLI, page ii
- Saving Changes to a Configuration, page xii
- Additional Information, page xii

For more information about using the CLI, see the "Using the Cisco IOS Command-Line Interface" section of the *Cisco IOS Configuration Fundamentals Configuration Guide*.

For information about the software documentation set, see the "About Cisco IOS and Cisco IOS XE Software Documentation" document.

Initially Configuring a Device

Initially configuring a device varies by platform. For information about performing an initial configuration, see the hardware installation documentation that is provided with the original packaging of the product or go to the Product Support area of Cisco.com at http://www.cisco.com/web/psa/products/index.html.

After you have performed the initial configuration and connected the device to your network, you can configure the device by using the console port or a remote access method, such as Telnet or Secure Shell (SSH), to access the CLI or by using the configuration method provided on the device, such as Security Device Manager.

Changing the Default Settings for a Console or AUX Port

There are only two changes that you can make to a console port and an AUX port:

- Change the port speed with the **config-register 0x** command. Changing the port speed is not recommended. The well-known default speed is 9600.
- Change the behavior of the port; for example, by adding a password or changing the timeout value.

Note The AUX port on the Route Processor (RP) installed in a Cisco ASR1000 series router does not serve any useful customer purpose and should be accessed only under the advisement of a customer support representative.

Using the CLI

This section describes the following topics:

- Understanding Command Modes, page ii
- Using the Interactive Help Feature, page v
- Understanding Command Syntax, page vi
- Understanding Enable and Enable Secret Passwords, page viii
- Using the Command History Feature, page viii
- Abbreviating Commands, page ix
- Using Aliases for CLI Commands, page ix
- Using the no and default Forms of Commands, page x
- Using the debug Command, page x
- Filtering Output Using Output Modifiers, page x
- Understanding CLI Error Messages, page xi

Understanding Command Modes

The CLI command mode structure is hierarchical, and each mode supports a set of specific commands. This section describes the most common of the many modes that exist.

Table 1 lists common command modes with associated CLI prompts, access and exit methods, and a brief description of how each mode is used.

Table 1 CLI Command Modes

Command Mode	**Access Method**	**Prompt**	**Exit Method**	**Mode Usage**
User EXEC	Log in.	`Router>`	Issue the **logout** or **exit** command.	• Change terminal settings. • Perform basic tests. • Display device status.
Privileged EXEC	From user EXEC mode, issue the **enable** command.	`Router#`	Issue the **disable** command or the **exit** command to return to user EXEC mode.	• Issue **show** and **debug** commands. • Copy images to the device. • Reload the device. • Manage device configuration files. • Manage device file systems.
Global configuration	From privileged EXEC mode, issue the **configure terminal** command.	`Router(config)#`	Issue the **exit** command or the **end** command to return to privileged EXEC mode.	Configure the device.
Interface configuration	From global configuration mode, issue the **interface** command.	`Router(config-if)#`	Issue the **exit** command to return to global configuration mode or the **end** command to return to privileged EXEC mode.	Configure individual interfaces.
Line configuration	From global configuration mode, issue the **line vty** or **line console** command.	`Router(config-line)#`	Issue the **exit** command to return to global configuration mode or the **end** command to return to privileged EXEC mode.	Configure individual terminal lines.

Table 1 CLI Command Modes (continued)

Command Mode	Access Method	Prompt	Exit Method	Mode Usage
ROM monitor	From privileged EXEC mode, issue the **reload** command. Press the **Break** key during the first 60 seconds while the system is booting.	`rommon # >` The # symbol represents the line number and increments at each prompt.	Issue the **continue** command.	• Run as the default operating mode when a valid image cannot be loaded. • Access the fall-back procedure for loading an image when the device lacks a valid image and cannot be booted. • Perform password recovery when a CTRL-Break sequence is issued within 60 seconds of a power-on or reload event.
Diagnostic (available only on the Cisco ASR1000 series router)	The router boots or enters diagnostic mode in the following scenarios. When a Cisco IOS process or processes fail, in most scenarios the router will reload. • A user-configured access policy was configured using the **transport-map** command, which directed the user into diagnostic mode. • The router was accessed using an RP auxiliary port. • A break signal (**Ctrl-C**, **Ctrl-Shift-6**, or the **send break** command) was entered, and the router was configured to enter diagnostic mode when the break signal was received.	`Router(diag)#`	If a Cisco IOS process failure is the reason for entering diagnostic mode, the failure must be resolved and the router must be rebooted to exit diagnostic mode. If the router is in diagnostic mode because of a transport-map configuration, access the router through another port or using a method that is configured to connect to the Cisco IOS CLI. If the RP auxiliary port was used to access the router, use another port for access. Accessing the router through the auxiliary port is not useful for customer purposes.	• Inspect various states on the router, including the Cisco IOS state. • Replace or roll back the configuration. • Provide methods of restarting the Cisco IOS software or other processes. • Reboot hardware, such as the entire router, an RP, an ESP, a SIP, a SPA, or possibly other hardware components. • Transfer files into or off of the router using remote access methods such as FTP, TFTP, and SCP.

EXEC commands are not saved when the software reboots. Commands that you issue in a configuration mode can be saved to the startup configuration. If you save the running configuration to the startup configuration, these commands will execute when the software is rebooted. Global configuration mode is the highest level of configuration mode. From global configuration mode, you can enter a variety of other configuration modes, including protocol-specific modes.

ROM monitor mode is a separate mode that is used when the software cannot load properly. If a valid software image is not found when the software boots or if the configuration file is corrupted at startup, the software might enter ROM monitor mode. Use the question symbol (?) to view the commands that you can use while the device is in ROM monitor mode.

```
rommon 1 > ?
alias                   set and display aliases command
boot                    boot up an external process
confreg                 configuration register utility
cont                    continue executing a downloaded image
context                 display the context of a loaded image
cookie                  display contents of cookie PROM in hex
.
.
.
rommon 2 >
```

The following example shows how the command prompt changes to indicate a different command mode:

```
Router> enable
Router# configure terminal
Router(config)# interface ethernet 1/1
Router(config-if)# ethernet
Router(config-line)# exit
Router(config)# end
Router#
```

Note A keyboard alternative to the **end** command is Ctrl-Z.

Using the Interactive Help Feature

The CLI includes an interactive Help feature. Table 2 describes how to use the Help feature.

Table 2 CLI Interactive Help Commands

Command	Purpose
help	Provides a brief description of the help feature in any command mode.
?	Lists all commands available for a particular command mode.
*partial command***?**	Provides a list of commands that begin with the character string (no space between the command and the question mark).
partial command<**Tab**>	Completes a partial command name (no space between the command and <Tab>).
command **?**	Lists the keywords, arguments, or both associated with the command (space between the command and the question mark).
command keyword **?**	Lists the arguments that are associated with the keyword (space between the keyword and the question mark).

The following examples show how to use the help commands:

help

```
Router> help

Help may be requested at any point in a command by entering a question mark '?'.  If
nothing matches, the help list will be empty and you must backup until entering a '?'
shows the available options.

Two styles of help are provided:

1. Full help is available when you are ready to enter a command argument (e.g. 'show ?')
and describes each possible argument.

2. Partial help is provided when an abbreviated argument is entered and you want to know
what arguments match the input (e.g. 'show pr?'.)
```

?

```
Router# ?
Exec commands:
  access-enable        Create a temporary access-List entry
  access-profile       Apply user-profile to interface
  access-template      Create a temporary access-List entry
  alps                 ALPS exec commands
  archive              manage archive files
<snip>
```

partial command?

```
Router(config)# zo?
zone  zone-pair
```

partial command<Tab>

```
Router(config)# we<Tab> webvpn
```

command ?

```
Router(config-if)# pppoe ?
  enable        Enable pppoe
  max-sessions  Maximum PPPOE sessions
```

command keyword ?

```
Router(config-if)# pppoe enable ?
  group  attach a BBA group
  <cr>
```

Understanding Command Syntax

Command syntax is the format in which a command should be entered in the CLI. Commands include the name of the command, keywords, and arguments. Keywords are alphanumeric strings that are used literally. Arguments are placeholders for values that a user must supply. Keywords and arguments may be required or optional.

Specific conventions convey information about syntax and command elements. Table 3 describes these conventions.

Table 3 CLI Syntax Conventions

Symbol/Text	Function	Notes
< > (angle brackets)	Indicate that the option is an argument.	Sometimes arguments are displayed without angle brackets.
A.B.C.D.	Indicates that you must enter a dotted decimal IP address.	Angle brackets (< >) are not always used to indicate that an IP address is an argument.
WORD (all capital letters)	Indicates that you must enter one word.	Angle brackets (< >) are not always used to indicate that a WORD is an argument.
LINE (all capital letters)	Indicates that you must enter more than one word.	Angle brackets (< >) are not always used to indicate that a LINE is an argument.
<cr> (carriage return)	Indicates the end of the list of available keywords and arguments, and also indicates when keywords and arguments are optional. When <cr> is the only option, you have reached the end of the branch or the end of the command if the command has only one branch.	—

The following examples show syntax conventions:

```
Router(config)# ethernet cfm domain ?
  WORD  domain name
Router(config)# ethernet cfm domain dname ?
  level
Router(config)# ethernet cfm domain dname level ?
  <0-7>  maintenance level number
Router(config)# ethernet cfm domain dname level 7 ?
  <cr>
Router(config)# snmp-server file-transfer access-group 10 ?
  protocol  protocol options
  <cr>
Router(config)# logging host ?
  Hostname or A.B.C.D  IP address of the syslog server
  ipv6                 Configure IPv6 syslog server
Router(config)# snmp-server file-transfer access-group 10 ?
  protocol  protocol options
  <cr>
```

Understanding Enable and Enable Secret Passwords

Some privileged EXEC commands are used for actions that impact the system, and it is recommended that you set a password for these commands to prevent unauthorized use. Two types of passwords, enable (not encrypted) and enable secret (encrypted), can be set. The following commands set these passwords and are issued in global configuration mode:

- **enable** *password*
- **enable secret** *password*

Using an enable secret password is recommended because it is encrypted and more secure than the enable password. When you use an enable secret password, text is encrypted (unreadable) before it is written to the config.text file. When you use an enable password, the text is written as entered (readable) to the config.text file.

Each type of password is case sensitive, can contain from 1 to 25 uppercase and lowercase alphanumeric characters, and can start with a number. Spaces are also valid password characters; for example, “two words” is a valid password. Leading spaces are ignored, but trailing spaces are recognized.

Note Both password commands have numeric keywords that are single integer values. If you choose a number for the first character of your password followed by a space, the system will read the number as if it were the numeric keyword and not as part of your password.

When both passwords are set, the enable secret password takes precedence over the enable password.

To remove a password, use the **no** form of the commands: **no enable** *password* or **no enable secret** *password*.

For more information about password recovery procedures for Cisco products, see http://www.cisco.com/en/US/products/sw/iosswrel/ps1831/products_tech_note09186a00801746e6.shtml.

Using the Command History Feature

The CLI command history feature saves the commands you enter during a session in a command history buffer. The default number of commands saved is 10, but the number is configurable within the range of 0 to 256. This command history feature is particularly useful for recalling long or complex commands.

To change the number of commands saved in the history buffer for a terminal session, issue the **terminal history size** command:

```
Router# terminal history size num
```

A command history buffer is also available in line configuration mode with the same default and configuration options. To set the command history buffer size for a terminal session in line configuration mode, issue the **history** command:

```
Router(config-line)# history [size num]
```

To recall commands from the history buffer, use the following methods:

- Press Ctrl-P or the up arrow key—Recalls commands beginning with the most recent command. Repeat the key sequence to recall successively older commands.

- Press Ctrl-N or the down arrow key—Recalls the most recent commands in the history buffer after they have been recalled using Ctrl-P or the up arrow key. Repeat the key sequence to recall successively more recent commands.

Note The arrow keys function only on ANSI-compatible terminals such as the VT100.

- Issue the **show history** command in user EXEC or privileged EXEC mode—Lists the most recent commands that you entered. The number of commands that are displayed is determined by the setting of the **terminal history size** and **history** commands.

  The CLI command history feature is enabled by default. To disable this feature for a terminal session, issue the **terminal no history** command in user EXEC or privileged EXEC mode or the **no history** command in line configuration mode.

Abbreviating Commands

Typing a complete command name is not always required for the command to execute. The CLI recognizes an abbreviated command when the abbreviation contains enough characters to uniquely identify the command. For example, the **show version** command can be abbreviated as **sh ver**. It cannot be abbreviated as **s ver** because **s** could mean **show**, **set**, or **systat**. The **sh v** abbreviation also is not valid because the **show** command has **vrrp** as a keyword in addition to **version**. (Command and keyword examples from Cisco IOS Release 12.4(13)T.)

Using Aliases for CLI Commands

To save time and the repetition of entering the same command multiple times, you can use a command alias. An alias can be configured to do anything that can be done at the command line, but an alias cannot move between modes, type in passwords, or perform any interactive functions.

Table 4 shows the default command aliases.

Table 4 Default Command Aliases

Command Alias	Original Command
h	help
lo	logout
p	ping
s	show
u or **un**	undebug
w	where

To create a command alias, issue the **alias** command in global configuration mode. The syntax of the command is **alias** *mode command-alias original-command*. Following are some examples:

- Router(config)# **alias exec prt partition**—privileged EXEC mode
- Router(config)# **alias configure sb source-bridge**—global configuration mode
- Router(config)# **alias interface rl rate-limit**—interface configuration mode

To view both default and user-created aliases, issue the **show alias** command.

For more information about the **alias** command, see
http://www.cisco.com/en/US/docs/ios/fundamentals/command/reference/cf_book.html.

Using the no and default Forms of Commands

Most configuration commands have a **no** form that is used to reset a command to its default value or disable a feature or function. For example, the **ip routing** command is enabled by default. To disable this command, you would issue the **no ip routing** command. To re-enable IP routing, you would issue the **ip routing** command.

Configuration commands may also have a **default** form, which returns the command settings to their default values. For commands that are disabled by default, using the **default** form has the same effect as using the **no** form of the command. For commands that are enabled by default and have default settings, the **default** form enables the command and returns the settings to their default values.

The **no** and **default** forms of commands are described in the command pages of command references.

Using the debug Command

A **debug** command produces extensive output that helps you troubleshoot problems in your network. These commands are available for many features and functions within Cisco IOS and Cisco IOS XE software. Some **debug** commands are **debug all**, **debug aaa accounting**, and **debug mpls packets**. To use **debug** commands during a Telnet session with a device, you must first enter the **terminal monitor** command. To turn off debugging completely, you must enter the **undebug all** command.

For more information about **debug** commands, see the *Cisco IOS Debug Command Reference* at
http://www.cisco.com/en/US/docs/ios/debug/command/reference/db_book.html.

⚠ **Caution** Debugging is a high priority and high CPU utilization process that can render your device unusable. Use **debug** commands only to troubleshoot specific problems. The best times to run debugging are during periods of low network traffic and when few users are interacting with the network. Debugging during these periods decreases the likelihood that the **debug** command processing overhead will affect network performance or user access or response times.

Filtering Output Using Output Modifiers

Many commands produce lengthy output that may use several screens to display. Using output modifiers, you can filter this output to show only the information that you want to see.

Three output modifiers are available and are described as follows:

- **begin** *regular expression*—Displays the first line in which a match of the regular expression is found and all lines that follow.
- **include** *regular expression*—Displays all lines in which a match of the regular expression is found.
- **exclude** *regular expression*—Displays all lines except those in which a match of the regular expression is found.

To use one of these output modifiers, type the command followed by the pipe symbol (|), the modifier, and the regular expression that you want to search for or filter. A regular expression is a case-sensitive alphanumeric pattern. It can be a single character or number, a phrase, or a more complex string.

The following example illustrates how to filter output of the **show interface** command to display only lines that include the expression "protocol."

```
Router# show interface | include protocol

FastEthernet0/0 is up, line protocol is up
Serial4/0 is up, line protocol is up
Serial4/1 is up, line protocol is up
Serial4/2 is administratively down, line protocol is down
Serial4/3 is administratively down, line protocol is down
```

Understanding CLI Error Messages

You may encounter some error messages while using the CLI. Table 5 shows the common CLI error messages.

Table 5 Common CLI Error Messages

Error Message	Meaning	How to Get Help
% Ambiguous command: "show con"	You did not enter enough characters for the command to be recognized.	Reenter the command followed by a space and a question mark (?). The keywords that you are allowed to enter for the command appear.
% Incomplete command.	You did not enter all the keywords or values required by the command.	Reenter the command followed by a space and a question mark (?). The keywords that you are allowed to enter for the command appear.
% Invalid input detected at "^" marker.	You entered the command incorrectly. The caret (^) marks the point of the error.	Enter a question mark (?) to display all the commands that are available in this command mode. The keywords that you are allowed to enter for the command appear.

For more system error messages, see the following documents:

- *Cisco IOS Release 12.2SR System Message Guide*
- *Cisco IOS System Messages, Volume 1 of 2* (Cisco IOS Release 12.4)
- *Cisco IOS System Messages, Volume 2 of 2* (Cisco IOS Release 12.4)

Saving Changes to a Configuration

To save changes that you made to the configuration of a device, you must issue the **copy running-config startup-config** command or the **copy system:running-config nvram:startup-config** command. When you issue these commands, the configuration changes that you made are saved to the startup configuration and saved when the software reloads or power to the device is turned off or interrupted. The following example shows the syntax of the **copy running-config startup-config** command:

```
Router# copy running-config startup-config
Destination filename [startup-config]?
```

You press Enter to accept the startup-config filename (the default), or type a new filename and then press Enter to accept that name. The following output is displayed indicating that the configuration was saved:

```
Building configuration...
[OK]
Router#
```

On most platforms, the configuration is saved to NVRAM. On platforms with a Class A flash file system, the configuration is saved to the location specified by the CONFIG_FILE environment variable. The CONFIG_FILE variable defaults to NVRAM.

Additional Information

- "Using the Cisco IOS Command-Line Interface" section of the *Cisco IOS Configuration Fundamentals Configuration Guide*:

  http://www.cisco.com/en/US/docs/ios/fundamentals/configuration/guide/cf_cli-basics.html

  or

  "Using Cisco IOS XE Software" chapter of the *Cisco ASR1000 Series Aggregation Services Routers Software Configuration Guide*:

  http://www.cisco.com/en/US/docs/routers/asr1000/configuration/guide/chassis/using_cli.html

- Cisco Product Support Resources

  http://www.cisco.com/web/psa/products/index.html

- Support area on Cisco.com (also search for documentation by task or product)

  http://www.cisco.com/en/US/support/index.html

- *White Paper: Cisco IOS Reference Guide*

  http://www.cisco.com/en/US/products/sw/iosswrel/ps1828/products_white_paper09186a008018305e.shtml

- Software Download Center (downloads; tools; licensing, registration, advisory, and general information) (requires Cisco.com User ID and password)

  http://www.cisco.com/kobayashi/sw-center/

- Error Message Decoder, a tool to help you research and resolve error messages for Cisco IOS software

  http://www.cisco.com/pcgi-bin/Support/Errordecoder/index.cgi

- Command Lookup Tool, a tool to help you find detailed descriptions of Cisco IOS commands (requires Cisco.com user ID and password)

  http://tools.cisco.com/Support/CLILookup

- Output Interpreter, a troubleshooting tool that analyzes command output of supported **show** commands

  https://www.cisco.com/pcgi-bin/Support/OutputInterpreter/home.pl\

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

Cisco IOS Optimized Edge Routing Feature Roadmap

First Published: January 29, 2007
Last Updated: July 11, 2008

This roadmap lists the features documented in the Cisco IOS Optimized Edge Routing configuration guide and maps them to the modules in which they appear.

Feature and Release Support

Table 1 lists Cisco IOS Optimized Edge Routing (OER) feature support for the following Cisco IOS software release trains:

- Cisco IOS Release 12.2SR
- Cisco IOS Release 12.2SX
- Cisco IOS Releases 12.3T, 12.4, and 12.4T

Only features that were introduced or modified in Cisco IOS Release 12.3(8)T, 12.2(33)SRB, 12.2(33)SXH, or a later release appear in the table. *Not all features may be supported in your Cisco IOS software release*.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Table 1 Supported Cisco IOS Optimized Edge Routing Features

Release	Feature Name	Feature Description	Where Documented
Cisco IOS Release 12.2SR			
12.2(33)SRB	OER BGP Inbound Optimization	This feature introduced support for best entrance selection for traffic that originates from prefixes outside an autonomous system destined for prefixes inside the autonomous system. External BGP (eBGP) advertisements from an autonomous system to another autonomous system (for example, an Internet service provider) can influence the entrance path for traffic entering the network. OER uses eBGP advertisements to manipulate the best entrance selection.	• "Cisco IOS Optimized Edge Routing Overview" • "Using OER to Profile the Traffic Classes" • "Measuring the Traffic Class Performance and Link Utilization Using OER" • "Configuring and Applying OER Policies" • "Using OER to Control Traffic Classes and Verify the Route Control Changes"
	OER DSCP Monitoring	This feature introduced automatic learning of traffic classes based on protocol, port numbers, and DSCP value. Traffic classes can be defined by a combination of keys comprising of protocol, port numbers, and DSCP values, with the ability to filter out traffic that is not required, and the ability to aggregate the traffic in which you are interested. Information such as protocol, port number, and DSCP information is now sent to the master controller database in addition to the prefix information. The new functionality allows OER to both actively and passively monitor application traffic.	• "Using OER to Profile the Traffic Classes" • "Measuring the Traffic Class Performance and Link Utilization Using OER" • "Configuring and Applying OER Policies" • "Using OER to Control Traffic Classes and Verify the Route Control Changes"

Table 1 Supported Cisco IOS Optimized Edge Routing Features (continued)

Release	Feature Name	Feature Description	Where Documented
12.2(33)SRB	OER Voice Traffic Optimization	This feature introduced support for outbound optimization of voice traffic based on the voice metrics, jitter and Mean Opinion Score (MOS). Jitter and MOS are important quantitative quality metrics for voice traffic and these voice metrics are measured using OER active probes.	• "OER Voice Traffic Optimization Using Active Probes" • "Measuring the Traffic Class Performance and Link Utilization Using OER" • "Configuring and Applying OER Policies" • "Using OER to Control Traffic Classes and Verify the Route Control Changes"
	OER Active Probe Source Address	This feature introduced the capability to configure a specific exit interface on the border router as the source for active probes.	"Measuring the Traffic Class Performance and Link Utilization Using OER"
	OER Application-Aware Routing: PBR	This feature introduced the capability to optimize IP traffic based on the type of application that is carried by the monitored prefix. Independent policy configuration is applied to the subset (application) of traffic.	• "Setting Up OER Network Components" • "Using OER to Profile the Traffic Classes" • "Configuring and Applying OER Policies" • "Using OER to Control Traffic Classes and Verify the Route Control Changes"

Table 1 Supported Cisco IOS Optimized Edge Routing Features (continued)

Release	Feature Name	Feature Description	Where Documented
12.2(33)SRB	OER Support for Cost-Based Optimization and Traceroute Reporting	This feature introduced the capability to configure exit link policies based on the ISP billing cost. This feature also introduces the capability to configure traceroute probes to determine prefix characteristics on a hop-by-hop basis.	• "Measuring the Traffic Class Performance and Link Utilization Using OER" • "Configuring and Applying OER Policies" • "Using OER to Control Traffic Classes and Verify the Route Control Changes"
	OER Support for Policy-Rules Configuration	This feature introduces the capability to select an OER map and apply the configuration under OER master controller configuration mode, providing an improved method to switch between predefined OER maps.	"Configuring and Applying OER Policies"
	Port and Protocol Based Prefix Learning	This feature introduced the capability to configure a master controller to learn prefixes based on the protocol type and the TCP or UDP port number.	"Using OER to Profile the Traffic Classes"
	VPN IPsec/GRE Tunnel Optimization	This module documents an OER solution that describes how to configure IP security (IPsec)/Generic Routing Encapsulation (GRE) tunnel interfaces as OER-managed exit links. Only network-based IPsec VPNs are supported.	"Configuring VPN IPsec/GRE Tunnel Interfaces As OER-Managed Exit Links"

Table 1 Supported Cisco IOS Optimized Edge Routing Features (continued)

Release	Feature Name	Feature Description	Where Documented
12.2(33)SRB	Optimized Edge Routing (OER)	OER provides automatic route optimization and load distribution for multiple connections between networks. OER is an integrated Cisco IOS solution that allows you to monitor IP traffic flows and then define policies and rules based on prefix performance, link load distribution, link bandwidth monetary cost, and traffic type. OER provides active and passive monitoring systems, dynamic failure detection, and automatic path correction. Deploying OER enables intelligent load distribution and optimal route selection in an enterprise network.	• "Cisco IOS Optimized Edge Routing Overview" • "Setting Up OER Network Components" • "Using OER to Profile the Traffic Classes" • "Measuring the Traffic Class Performance and Link Utilization Using OER" • "Configuring and Applying OER Policies" • "Using OER to Control Traffic Classes and Verify the Route Control Changes"
Cisco IOS Release 12.2SX			
12.2(33)SXH	OER Border Router Only Functionality	In Cisco IOS Release 12.2(33)SXH support for using a Cisco Catalyst 6500 series switch as an OER border router was introduced. Only border router functionality is included in the Cisco IOS Release 12.2(33)SXH images; no master controller configuration is available. The master controller that communicates with the Cisco Catalyst 6500 series switch being used as a border router must be a router running Cisco IOS Release 12.4(6)T or a later release. The OER master controller software has been modified to handle the limited functionality supported by the Cisco Catalyst 6500 border routers. Using the Route Processor (RP), the Catalyst 6500 border routers can capture throughput statistics only for a traffic class compared to the delay, loss, unreachability, and throughput statistics collected by non-Catalyst 6500 border routers. A master controller automatically detects the limited capabilities of the Catalyst 6500 border routers and downgrades other border routers to capture only the throughput statistics for traffic classes. By ignoring other types of statistics, the master controller is presented with a uniform view of the border router functionality.	• "Setting Up OER Network Components" • "Using OER to Profile the Traffic Classes" • "Measuring the Traffic Class Performance and Link Utilization Using OER"

Table 1 Supported Cisco IOS Optimized Edge Routing Features (continued)

Release	Feature Name	Feature Description	Where Documented
Cisco IOS Releases 12.3T, 12.4, and 12.4T			
12.4(20)T	Performance Routing with NBAR/CCE Application Recognition	The Performance Routing with NBAR/CCE Application Recognition feature introduces the ability to profile an application-based traffic class using NBAR. Network-Based Application Recognition (NBAR) is a classification engine that recognizes and classifies a wide variety of protocols and applications, including web-based and other difficult-to-classify applications and protocols that use dynamic TCP/UDP port assignments. PfR uses NBAR to recognize and classify a protocol or application, and the resulting traffic classes are added to the PfR application database to be passively and actively monitored.	"Using Performance Routing to Profile the Traffic Classes"
12.4(15)T	OER - Application Aware Routing with Static Application Mapping	This feature introduces the ability to configure standard applications using just one keyword. In Cisco IOS Release 12.4(9)T, and prior releases, the definition of application traffic involves some awkward configuration. This feature also introduces a learn list configuration mode that allows Optimized Edge Routing (OER) policies to be applied to traffic classes profiled in a learn list. Different policies can be applied to each learn list. New **traffic-class** and **match traffic-class** commands are introduced to simplify the configuration of traffic classes that OER can automatically learn, or that can be manually configured.	"Using OER to Profile the Traffic Classes"
	Performance Routing - Application Interface	This feature introduces support for an OER application interface. The application interface defines the mode of communication and messaging between applications and the network for the purpose of optimizing the traffic associated with the applications. A provider must be registered with an OER master controller before the application can interface with OER. Host devices in the provider network running an application that communicates with OER using the application interface must also be configured at an OER master controller with an IP address and key chain password.	"Setting Up OER Network Components"
	Performance Routing - Link Groups	This feature introduces the ability to define a group of exit links as a preferred set of links, or a fallback set of links for OER to use when optimizing traffic classes specified in an OER policy.	"Configuring and Applying OER Policies"

Table 1 Supported Cisco IOS Optimized Edge Routing Features (continued)

Release	Feature Name	Feature Description	Where Documented
12.4(9)T	OER BGP Inbound Optimization	This feature introduced support for best entrance selection for traffic that originates from prefixes outside an autonomous system destined for prefixes inside the autonomous system. External BGP (eBGP) advertisements from an autonomous system to another autonomous system (for example, an Internet service provider) can influence the entrance path for traffic entering the network. OER uses eBGP advertisements to manipulate the best entrance selection.	• "Cisco IOS Optimized Edge Routing Overview" • "Using OER to Profile the Traffic Classes" • "Measuring the Traffic Class Performance and Link Utilization Using OER" • "Configuring and Applying OER Policies" • "Using OER to Control Traffic Classes and Verify the Route Control Changes"
	OER DSCP Monitoring	This feature introduced automatic learning of traffic classes based on protocol, port numbers, and DSCP value. Traffic classes can be defined by a combination of keys comprising of protocol, port numbers, and DSCP values, with the ability to filter out traffic that is not required, and the ability to aggregate the traffic in which you are interested. Information such as protocol, port number, and DSCP information is now sent to the master controller database in addition to the prefix information. The new functionality allows OER to both actively and passively monitor application traffic.	• "Using OER to Profile the Traffic Classes" • "Measuring the Traffic Class Performance and Link Utilization Using OER" • "Configuring and Applying OER Policies" • "Using OER to Control Traffic Classes and Verify the Route Control Changes"

Table 1 ***Supported Cisco IOS Optimized Edge Routing Features (continued)***

Release	Feature Name	Feature Description	Where Documented
12.4(6)T	OER Voice Traffic Optimization	This feature introduced support for outbound optimization of voice traffic based on the voice metrics, jitter and Mean Opinion Score (MOS). Jitter and MOS are important quantitative quality metrics for voice traffic and these voice metrics are measured using OER active probes.	• "OER Voice Traffic Optimization Using Active Probes" • "Measuring the Traffic Class Performance and Link Utilization Using OER" • "Configuring and Applying OER Policies" • "Using OER to Control Traffic Classes and Verify the Route Control Changes"
12.4(2)T	OER Active Probe Source Address	This feature introduced the capability to configure a specific exit interface on the border router as the source for active probes.	"Measuring the Traffic Class Performance and Link Utilization Using OER"
	OER Application-Aware Routing: PBR	This feature introduced the capability to optimize IP traffic based on the type of application that is carried by the monitored prefix. Independent policy configuration is applied to the subset (application) of traffic.	• "Setting Up OER Network Components" • "Using OER to Profile the Traffic Classes" • "Configuring and Applying OER Policies" • "Using OER to Control Traffic Classes and Verify the Route Control Changes"

Table 1 Supported Cisco IOS Optimized Edge Routing Features (continued)

Release	Feature Name	Feature Description	Where Documented
12.3(14)T	OER Support for Cost-Based Optimization and Traceroute Reporting	This feature introduced the capability to configure exit link policies based on the ISP billing cost. This feature also introduces the capability to configure traceroute probes to determine prefix characteristics on a hop-by-hop basis.	• "Measuring the Traffic Class Performance and Link Utilization Using OER" • "Configuring and Applying OER Policies" • "Using OER to Control Traffic Classes and Verify the Route Control Changes"

Table 1 ***Supported Cisco IOS Optimized Edge Routing Features (continued)***

Release	Feature Name	Feature Description	Where Documented
12.3(11)T	OER Support for Policy-Rules Configuration	This feature introduces the capability to select an OER map and apply the configuration under OER master controller configuration mode, providing an improved method to switch between predefined OER maps.	"Configuring and Applying OER Policies"
	Port and Protocol Based Prefix Learning	This feature introduced the capability to configure a master controller to learn prefixes based on the protocol type and the TCP or UDP port number.	"Using OER to Profile the Traffic Classes"
	VPN IPsec/GRE Tunnel Optimization	This module documents an OER solution that describes how to configure IP security (IPsec)/Generic Routing Encapsulation (GRE) tunnel interfaces as OER-managed exit links. Only network-based IPsec VPNs are supported.	"Configuring VPN IPsec/GRE Tunnel Interfaces As OER-Managed Exit Links"
12.3(8)T	Optimized Edge Routing (OER)	OER provides automatic route optimization and load distribution for multiple connections between networks. OER is an integrated Cisco IOS solution that allows you to monitor IP traffic flows and then define policies and rules based on prefix performance, link load distribution, link bandwidth monetary cost, and traffic type. OER provides active and passive monitoring systems, dynamic failure detection, and automatic path correction. Deploying OER enables intelligent load distribution and optimal route selection in an enterprise network.	• "Cisco IOS Optimized Edge Routing Overview" • "Setting Up OER Network Components" • "Using OER to Profile the Traffic Classes" • "Measuring the Traffic Class Performance and Link Utilization Using OER" • "Configuring and Applying OER Policies" • "Using OER to Control Traffic Classes and Verify the Route Control Changes"

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

Cisco IOS Optimized Edge Routing Overview

First Published: January 29, 2007
Last Updated: February 28, 2007

Optimized Edge Routing (OER) provides automatic route optimization and load distribution for multiple connections between networks. OER is an integrated Cisco IOS solution that allows you to monitor IP traffic flows and then define policies and rules based on traffic class performance, link load distribution, link bandwidth monetary cost, and traffic type. OER provides active and passive monitoring systems, dynamic failure detection, and automatic path correction. Deploying OER enables intelligent load distribution and optimal route selection in an enterprise network.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Optimized Edge Routing Overview" section on page 8.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Information About Optimized Edge Routing

To configure OER, you should understand the following concepts:

- OER Overview, page 2
- OER Network Performance Loop, page 2
- OER and the Enterprise Network, page 5

OER Overview

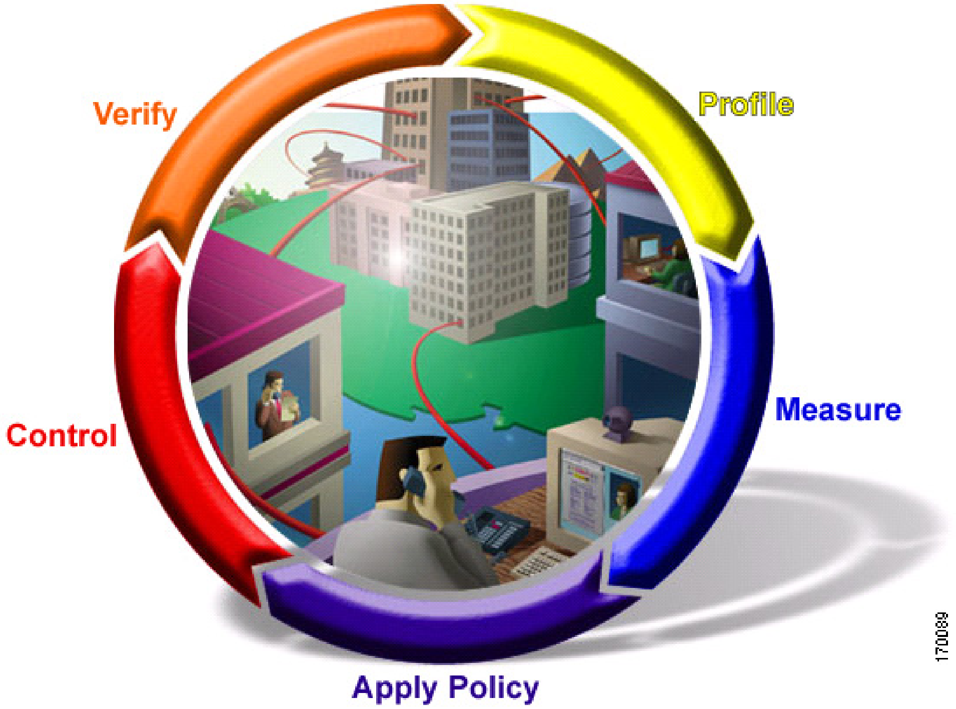
OER was developed to identify and control network performance issues that traditional IP routing cannot address. In traditional IP routing, each peer device communicates its view of reachability to a prefix destination with some concept of a cost related to reaching the metric. The best path route to a prefix destination is usually determined using the least cost metric, and this route is entered into the routing information base (RIB) for the device. As a result, any route introduced into the RIB is treated as the best path to control traffic destined for the prefix destination. The cost metric is configured to reflect a statically engineered view of the network, for example, the cost metric is a reflection of either a user preference for a path or a preference for a higher bandwidth interface (inferred from the type of interface). The cost metric does not reflect the state of the network or the state of the performance of traffic traveling on that network at that time. Traditional IP routed networks are therefore adaptive to physical state changes in the network (for example, interfaces going down) but not to performance changes (degradation or improvement) in the network. Occasionally, degradation in traffic can be inferred from either the degradation in performance of the routing device or the loss of session connectivity, but these traffic degradation symptoms are not a direct measure of the performance of the traffic and cannot be used to influence decisions about best-path routing.

To address performance issues for traffic within a network, OER manages traffic classes. Traffic classes are defined as subsets of the traffic on the network, and a subset may represent the traffic associated with an application, for example. The performance of each traffic class is measured and compared against configured or default metrics defined in an OER policy. OER monitors the traffic class performance and selects the best entrance or exit for the traffic class. If the subsequent traffic class performance does not conform to the policy, OER selects another entrance or exit for the traffic class.

OER Network Performance Loop

Every traditional routing protocol creates a feedback loop among devices to create a routing topology. OER infrastructure includes a performance routing protocol that is communicated in a client-server messaging mode. The routing protocol employed by OER runs between a network controller called a master controller and performance-aware devices called border routers. This performance routing protocol creates a network performance loop in which the network profiles which traffic classes have to be optimized, measures and monitors the performance metrics of the identified traffic classes, applies policies to the traffic classes, and routes the identified traffic classes based on the best performance path. Figure 1 shows the five OER phases: profile, measure, apply policy, control, and verify.

Figure 1 OER Network Performance Loop



To introduce OER in your network you should understand and implement the following five OER phases:

- OER Profile Phase, page 3
- OER Measure Phase, page 4
- OER Apply Policy Phase, page 4
- OER Control Phase, page 4
- OER Verify Phase, page 5

The OER performance loop starts with the profile phase followed by the measure, apply policy, control, and verify phases. The flow continues after the verify phase back to the profile phase to update the traffic classes and cycle through the process.

OER Profile Phase

In medium to large networks there are hundreds of thousands of routes in the RIB to which a device is trying to route traffic. Because performance routing is a means of preferring some traffic over another, a subset of the total routes in the RIB has to be selected to optimize for performance routing. In the OER profile phase this selection of the subset of total traffic flowing through a device or a network is accomplished in a combination of ways:

- The device profiles the traffic that has to be performance routed (optimized) by learning the flows that pass through the device and by selecting those flows that have the highest delay or the highest throughput.
- In addition to, or instead of learning, you can configure a class of traffic to performance route.

OER Measure Phase

After you have profiled a group of traffic classes that are to be performance routed, the network has to measure the performance metrics of these individual traffic classes. There are two mechanisms—passive monitoring and active monitoring—to measure performance metrics, and one or both could be deployed in the network to accomplish this task. Monitoring is the act of measuring at periodic intervals.

Passive monitoring is the act of measuring the performance metrics of the traffic flow as the flow is traversing the device in the data path. Passive monitoring cannot be employed for measuring performance metrics for some traffic classes, and there are some hardware or software limitations.

Active monitoring consists of generating synthetic traffic to emulate the traffic class that is being monitored. The synthetic traffic is measured instead of the actual traffic class. Then the results of the synthetic traffic monitoring are applied to performance route the traffic class represented by the synthetic traffic.

You can also deploy both passive and active monitoring modes in an automated flow. The passive monitoring phase may detect traffic class performance that does not conform to an OER policy, and then active monitoring can be applied to that traffic class to find the best alternate performance path, if available. For more details about OER policies, see the "OER Apply Policy Phase" section.

OER Apply Policy Phase

After collecting the performance metrics of the class of traffic that you want to optimize, OER compares the results with a set of configured low and high thresholds for each metric. When a metric, and consequently a policy, goes out of bounds, it is an Out-of-Policy (OOP) event. The results are compared on a relative basis—a deviation from the observed mean—or on a threshold basis—the lower or upper bounds of a value—or a combination of both.

There are two types of policies that can be defined in OER: traffic class policies and link policies. Traffic class policies are defined for prefixes or for applications. Link policies are defined for exit or entrance links at the network edge. Both types of OER policies define the criteria for determining an OOP event. The policies are applied on a global basis in which a set of policies is applied to all traffic classes, or on a more targeted basis in which a set of policies is applied to a selected (filtered) list of traffic classes.

With multiple policies, many performance metric parameters, and different ways of assigning these policies to traffic classes, a method of resolving policy conflicts was created. The default arbitration method uses a default priority level given to each performance metric variable and each policy. You can configure different priority levels that overrides the default arbitration for all policies, or a selected set of policies.

OER Control Phase

In the OER control phase (also called the enforce phase) of the performance loop, the traffic is controlled to enhance the network performance time. The technique used to control the traffic depends on the class of traffic. For traffic classes that are defined using a prefix only, the prefix reachability information used in traditional routing can be manipulated. Protocols such as Border Gateway Protocol (BGP) or RIP are used to announce or remove the prefix reachability information by introducing or deleting a route and its appropriate cost metrics.

For traffic classes that are defined by an application in which a prefix and additional packet matching criteria are specified, OER cannot employ traditional routing protocols because routing protocols communicate the reachability of the prefix only. For these application traffic classes, OER uses two control methods: device specific and network specific.

The device specific control method is achieved using interaction with policy-based routing (PBR).

The network specific control method is achieved in two ways:

- Overlay Performance Networks—An overlay network is created where each device at the edge of the network can learn about the existence of every other device at the network edge. Multiprotocol Label Switching (MPLS), or multipoint Generic Routing Encapsulation (mGRE) technology can then be used to reach the required edge device.
- Context Enhanced Protocols—The existing routing protocols (BGP, Open Shortest Path First (OSPF), Enhanced Interior Gateway Routing Protocol (EIGRP) are enhanced to communicate information about a context attached to a prefix. The extra packet matching criteria in the traffic class flow would make the context which is attached to the prefix in their route updates.

OER Verify Phase

During the OER control phase if a traffic class is OOP, then OER introduces controls to influence (optimize) the flow of the traffic for the traffic class that is OOP. A static route and a BGP route are examples of controls introduced by OER into the network. After the controls are introduced, OER will verify that the optimized traffic is flowing through the preferred exit or entrance links at the network edge. If the traffic class remains OOP, OER will drop the controls that were introduced to optimize the traffic for the OOP traffic class and cycle through the network performance loop.

OER and the Enterprise Network

Enterprise networks use multiple Internet Service Provider (ISP) or WAN connections at the network edge for reliability and load distribution. Existing reliability mechanisms depend on link state or route removal on the border router to select the best exit link for a prefix or set of prefixes. Multiple connections protect enterprise networks from catastrophic failures but do not protect the network from brownouts, or soft failures, that occur because of network congestion. Existing mechanisms can respond to catastrophic failures at the first indication of a problem. However, blackouts and brownouts can go undetected and often require the network operator to take action to resolve the problem. When a packet is transmitted between external networks (nationally or globally), the packet spends the vast majority of its life cycle on the WAN segments of the network. Optimizing WAN route selection in the enterprise network provides the end-user with the greatest performance improvement, even better than LAN speed improvements in the local network.

Although many of the examples used to describe OER deployment show ISPs as the network with which the edge devices communicate, there are other solutions. The network edge can be defined as any logical separation in a network: can be another part of the network such as a data center network within the same location, as well as WAN and ISP connections. The network, or part of the network, connected to the original network edge devices must have a separate autonomous system number when communicating using BGP.

OER is implemented in Cisco IOS software as an integrated part of Cisco core routing functionality. Deploying OER enables intelligent network traffic load distribution and dynamic failure detection for data paths at the network edge. While other routing mechanisms can provide both load distribution and failure mitigation, only OER can make routing adjustments based on criteria other than static routing metrics, such as response time, packet loss, path availability, and traffic load distribution. Deploying OER allows you to optimize network performance and link load utilization while minimizing bandwidth costs and reducing operational expenses.

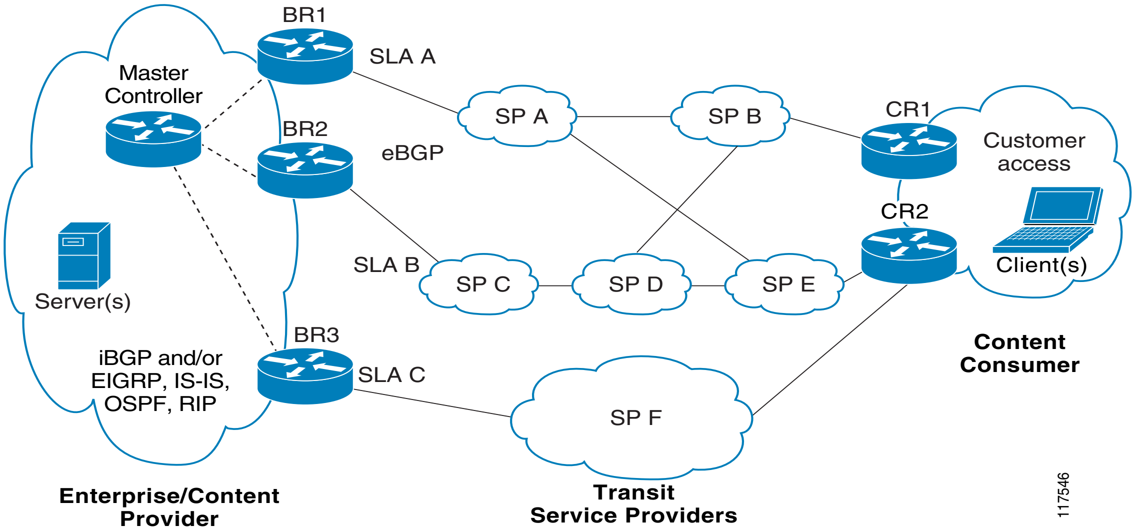
The following two sections give an overview of a typical deployment of OER and the network components managed by OER:

- Typical Deployment of OER, page 6
- Network Components Managed by OER, page 6

Typical Deployment of OER

Figure 2 shows a typical OER-managed enterprise network of a content provider. The enterprise network has three exit interfaces that are used to deliver content to customer access networks. The content provider has a separate service level agreement (SLA) with a different ISP for each exit link. The customer access network has two edge routers that connect to the Internet. Traffic is carried between the enterprise network and the customer access network over six service provider (SP) networks.

Figure 2 A Typical OER Deployment



OER monitors and controls outbound traffic on the three border routers (BRs). In Cisco IOS Release 12.4(9)T, the ability to monitor and control inbound traffic was introduced. OER measures the packet response time and path availability from the egress interfaces on BR1, BR2 and BR3. Changes to exit link performance on the border routers are detected on a per-prefix basis. If the performance of a prefix falls below default or user-defined policy parameters, routing is altered locally in the enterprise network to optimize performance and to route around failure conditions that occur outside of the enterprise network. For example, an interface failure or network misconfiguration in the SP D network can cause outbound traffic that is carried over the BR2 exit interface to become congested or fail to reach the customer access network. Traditional routing mechanisms cannot anticipate or resolve these types of problems without intervention by the network operator. OER can detect failure conditions and automatically alter routing inside of the network to compensate.

Network Components Managed by OER

OER is configured on Cisco routers using Cisco IOS command-line interface (CLI) configurations. An OER deployment has two primary components, a master controller and one or more border routers. The master controller is the intelligent decision maker, while the border routers are enterprise edge routers with exit interfaces that are either used to access the Internet or used as WAN exit links.

OER Master Controller

The master controller is a single router that coordinates all OER functions within an OER-managed network. A Cisco router can be configured to run a standalone master controller process or can also be configured to perform other functions, such as routing or running a border router process. The master controller maintains communication and authenticates the sessions with the border routers. The master controller monitors outbound traffic flows using active or passive monitoring and then applies default or user-defined policies to alter routing to optimize prefixes and exit links. OER administration and control is centralized on the master controller, which makes all policy decisions and controls the border routers.

OER Border Router

The border router is an enterprise edge router with one or more exit links to an ISP or other participating network. The border router is where all policy decisions and changes to routing in the network are enforced. The border router participates in prefix monitoring and route optimization by reporting prefix and exit link measurements to the master controller and then by enforcing policy changes received from the master controller. The border router enforces policy changes by injecting a preferred route to alter routing in the network. The border router is deployed on the edge of the network, so the border router must be in the forwarding path. A border router process can be enabled on the same router as a master controller process.

OER-Managed Network Interfaces

An OER-managed network must have at least two egress interfaces that can carry outbound traffic and can be configured as external interfaces. These interfaces should connect to an ISP or WAN link (Frame-Relay, ATM) at the network edge. The router must also have one interface (reachable by the internal network) that can be configured as an internal interface for passive monitoring. There are three interface configurations required to deploy OER: external interfaces, internal interfaces, and local interfaces.

For more details about the master controller, border routers, and interfaces used by OER, see the "Setting Up OER Network Components" module.

Where to Go Next

If this is the first time you have read this document and you are ready to implement OER in your network, proceed to the "Setting Up OER Network Components" module. If you have set up your OER components, you should read through the other modules in the following order:

- Using OER to Profile the Traffic Classes
- Measuring the Traffic Class Performance and Link Utilization Using OER
- Configuring and Applying OER Policies
- Using OER to Control Traffic Classes and Verify the Route Control Changes

After you understand the various OER phases, review the OER solutions modules that are listed under "Related Documents" section on page 8.

Additional References

The following sections provide references related to Cisco IOS Optimized Edge Routing Overview.

Related Documents

Related Topic	Document Title
Location of OER features	"Cisco IOS OER Features Roadmap" module
OER solution module: voice traffic optimization using OER active probes.	"OER Voice Traffic Optimization Using Active Probes" module
OER solution module: configuring VPN IPsec/GRE tunnel interfaces as OER-managed exit links.	"Configuring VPN IPsec/GRE Tunnel Interfaces As OER-Managed Exit Links" module
Cisco OER commands: complete command syntax, command mode, command history, defaults, usage guidelines and examples	*Cisco IOS Optimized Edge Routing Command Reference*
IP SLAs overview	*Cisco IOS IP SLAs Overview*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Optimized Edge Routing Overview

Table 1 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Releases 12.3(8)T, 12.2(33)SRB, or a later release appear in the table.

For information on a feature in this technology that is not documented here, see the "Cisco IOS Optimized Edge Routing Features Roadmap."

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Optimized Edge Routing Overview

Feature Name	Releases	Feature Information
Optimized Edge Routing	12.3(8)T 12.2(33)SRB	OER was introduced.
OER BGP Inbound Optimization	12.4(9)T 12.2(33)SRB	OER Border Gateway Protocol (BGP) inbound optimization supports best entrance selection for traffic that originates from prefixes outside an autonomous system destined for prefixes inside the autonomous system. External BGP (eBGP) advertisements from an autonomous system to another autonomous system (for example, an Internet service provider) can influence the entrance path for traffic entering the network. OER uses eBGP advertisements to manipulate the best entrance selection. The following section provides information about this feature: • Typical Deployment of OER, page 6 The following commands were introduced or modified by this feature: **clear oer master prefix**, **downgrade bgp**, **inside bgp**, **match ip address (OER)**, **match oer learn**, **max range receive**, **max utilization receive**, **show oer master prefix**.

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

Setting Up OER Network Components

First Published: January 29, 2007
Last Updated: April 10, 2008

This module describes the concepts and tasks to help you set up the network components required for an Optimized Edge Routing (OER)-managed network. OER network components are described and configuration tasks are provided to help you configure a master controller (MC) and one or more border routers (BRs) that enable communication between these two software components.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Setting Up OER Network Components" section on page 59.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Setting Up OER Network Components

- Before setting up OER network components, you should be familiar with the "Cisco IOS Optimized Edge Routing Overview" module.
- Cisco Express Forwarding (CEF) must be enabled on all participating routers.
- Either routing protocol peering must be established on your network or static routing must be configured before setting up OER network components.

  If you have configured internal Border Gateway Protocol (iBGP) on the border routers, BGP peering must be either established and consistently applied throughout your network or redistributed into the Interior Gateway Protocol (IGP).

  If an IGP is deployed in your network, static route redistribution must be configured with the **redistribute** command. IGP or static routing should also be applied consistently throughout an OER-managed network; the border router should have a consistent view of the network.

Restrictions for Setting Up OER Network Components

- OER supports only IP security (IPsec) or Generic Routing Encapsulation (GRE) Virtual Private Networks (VPNs). No other VPN types are supported.
- When two or more border routers are deployed in an OER-managed network, the next hop on each border router, as installed in the Routing Information Base (RIB), cannot be an address from the same subnet as the next hop on the other border router.
- Interfaces that are configured to be under OER control can also carry multicast traffic. However, if the source of the multicast traffic comes from outside of the OER-managed network and inbound multicast traffic is carried over OER-managed exit links, the source multicast address should be excluded from OER control.
- Internet exchange points where a border router can communicate with several service providers over the same broadcast media are not supported.
- Token Ring interfaces are not supported by OER and cannot be configured as OER-managed interfaces. It may be possible to load a Token Ring interface configuration under certain conditions. However, the Token Ring interface will not become active and the border router will not function if the Token Ring interface is the only external interface on the border router.

Information About Setting Up OER Network Components

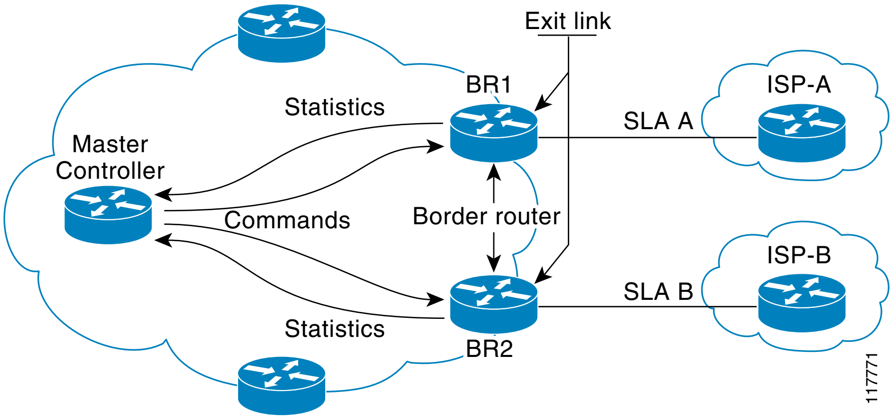
To configure a basic OER-managed network, you should understand the following concepts:

- OER Application Interface, page 12
- OER Logging and Reporting, page 14

OER-Managed Network

Figure 1 shows an OER-managed network. This network contains a master controller and two border routers. OER is configured on Cisco routers using the Cisco IOS command-line interface (CLI). OER deployment has two primary components: a master controller and one or more border routers. The master controller is the intelligent decision maker, while the border routers are enterprise edge routers with exit interfaces at the network edge. Border routers are either used to access the Internet or used as WAN exit links. OER communication between the master controller and the border routers is carried separately from routing protocol traffic. This communication is protected by Message Digest 5 (MD5) authentication. Each border router has both an external interface, which is connected, for example, to an ISP by a WAN link, and an internal interface that is reachable by the master controller.

Figure 1 OER-Managed Network

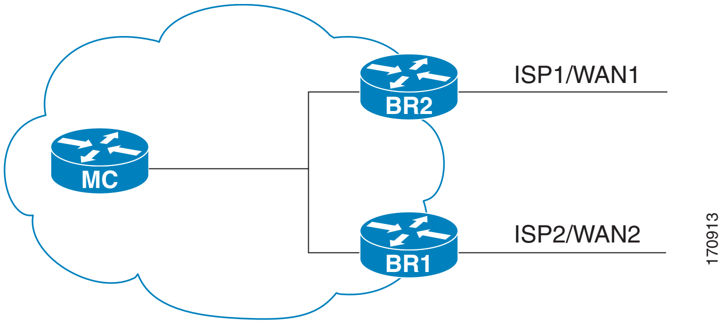


External interfaces are used to forward outbound traffic from the network and as the source for active monitoring. Internal interfaces are used for OER communication and for passive monitoring. In Cisco IOS Release 12.4(9)T, the ability to monitor and control inbound traffic was introduced. At least one external and one internal interface must be configured on each border router. At least two external interfaces are required in an OER-managed network. A local interface is configured on the border router for communication with the master controller.

OER Master Controller

The master controller is a single router that coordinates all OER functions within an OER-managed network. A Cisco router can be configured either to run a standalone master controller process or to perform other functions, such as routing or running a border router process. Figure 2 shows an example of a standalone router configured as a master controller.

Figure 2 Master Controller Example



The master controller maintains communication and authenticates the sessions with the border routers. Outbound traffic flows are monitored by the border routers using active or passive monitoring, and the data is collected in a central policy database residing on the router configured as the master controller. Then the master controller applies default or user-defined policies to alter routing to optimize prefixes and exit links. OER administration and control is centralized on the master controller, which makes all policy decisions and controls the border routers. The master controller does not have to be in the traffic forwarding path, but it must be reachable by the border routers. The master controller can support up to 10 border routers and up to 20 OER-managed external interfaces.

Central Policy Database

The master controller continuously monitors the network and maintains a central policy database in which collected statistical information is stored. The master controller compares long-term and short-term measurements. The long-term measurements are collected every 60 minutes. Short-term measurements are collected every 5 minutes. The master controller analyzes these statistics to determine which routes have the lowest delay, highest outbound throughput, relative or absolute packet loss, relative or absolute link cost, and prefix reachability to analyze and optimize the performance of monitored prefixes and to distribute the load from overutilized exit links to underutilized exit links. The locations of the exit links on the border routers are shown in Figure 1.

Tip We recommend that the master controller be physically close to the border routers to minimize communication response time in OER-managed networks. If traffic is to be routed between border routers, the border routers also should be physically close each other to minimize the number of hops.

Border Routers in an OER-Managed Network

The border router is an enterprise edge router with one or more exit links to another participating network, such as an Internet Service Provider (ISP), and is the site where all policy decisions and changes to routing in the network are enforced. The border router participates in prefix monitoring and route optimization by first reporting prefix and exit link measurements to the master controller and then by enforcing policy changes received from the master controller. The border router enforces policy changes by injecting a preferred route to alter routing in the network. The border router is deployed on the edge of the network, so the border router must be in the forwarding path. A border router process can be enabled on the same router as a master controller process.

Policy Enforcement Point

The border router is the policy enforcement point. Default or user-defined policies are configured on the master controller to set the performance level for prefixes and exit links. The master controller automatically alters routing in the OER-managed network, as necessary, by sending control commands to the border routers to inject a preferred route. The preferred route is advertised or redistributed through the internal network. The preferred route alters default routing behavior so that out-of-policy prefixes are moved from overutilized exit links to underutilized exit links, bringing prefixes and exit links in-policy, thus optimizing the overall performance of the enterprise network.

Tip We recommend that if traffic is to be routed between border routers, the border routers should be physically close to each other to minimize the number of hops. The master controller also should be physically close to the border routers to minimize communication response time in OER-managed networks.

Single Hop Peer Restriction Avoidance using OER Interim Border Routers

In Cisco IOS Release 12.4(2)T and 12.2(33)SRB, support for a border router that is more than one hop away from another border router was introduced. In releases prior to Cisco IOS Release 12.4(2)T, the border routers must be one hop away from each other. However, if the design of your network requires the border routers to be separated by more than one hop, a Cisco router between the border routers can be configured as an interim border router. The interim border routers act as transit routers between the border routers in your network. The master controller discovers the paths between interim and standard border routers and policy routes traffic through the appropriate external interface on a standard border router.

The configuration of an interim border router is similar to standard border router configuration. There is only one exception. No external interfaces are defined in the master controller configuration for the interim border router. However, a single internal interface must be configured for the interim border router to establish connectivity with the master controller. The configuration on the interim border router is the same as with a standard border router.

Note Multihop border router peerings are not supported.

OER Border Router Support for Cisco Catalyst 6500 Series Switches

In Cisco IOS Release 12.2(33)SXH support for using a Cisco Catalyst 6500 series switch as an OER border router was introduced. Only border router functionality is included in the Cisco IOS Release 12.2(33)SXH images; no master controller configuration is available. The master controller that communicates with the Cisco Catalyst 6500 series switch being used as a border router must be a router running Cisco IOS Release 12.4(6)T or a later release.

The OER master controller software has been modified to handle the limited functionality supported by the Cisco Catalyst 6500 border routers. Using the Route Processor (RP), the Catalyst 6500 border routers can capture throughput statistics only for a traffic class compared to the delay, loss, unreachability, and throughput statistics collected by non-Catalyst 6500 border routers. A master controller will automatically detect the limited capabilities of the Catalyst 6500 border routers and will downgrade other border routers to capture only the throughput statistics for traffic classes. By ignoring other types of statistics, the master controller is presented with a uniform view of the border router functionality.

If one of the border router is identified as a Catalyst 6500 border router, then the master controller starts periodic active probing of the all the traffic classes under OER control and ignores the passive performance statistics. Active probing results received for each traffic class are evaluated against the policies configured for that traffic class.

For more details about profiling and monitoring modifications introduced to support the Cisco Catalyst 6500 series switch as an OER border router, see the "Measuring the Traffic Class Performance and Link Utilization Using OER" module and the "Using OER to Profile the Traffic Classes" module.

OER-Managed Network Interfaces

An OER-managed network must have at least two egress interfaces that can carry outbound traffic and that can be configured as external interfaces. These interfaces should connect to an ISP or WAN link (Frame-Relay, ATM) at the network edge. The router must also have one interface (reachable by the internal network) that can be configured as an internal interface for passive monitoring. There are three interface configurations required to deploy OER:

- *External interfaces* are configured as OER-managed exit links to forward traffic. The physical external interface is enabled on the border router. The external interface is configured as an OER external interface on the master controller. The master controller actively monitors prefix and exit link performance on these interfaces. Each border router must have at least one external interface, and a minimum of two external interfaces are required in an OER-managed network.
- *Internal interfaces* are used only for passive performance monitoring with NetFlow. No explicit NetFlow configuration is required. The internal interface is an active border router interface that connects to the internal network. The internal interface is configured as an OER-internal interface on the master controller. At least one internal interface must be configured on each border router.
- *Local interfaces* are used only for master controller and border router communication. A single interface must be configured as a local interface on each border router. The local interface is identified as the source interface for communication with the master controller.

Tip If a master controller and border router process are enabled on the same router, a loopback interface should be configured as the local interface.

The following interface types can be configured as external and internal interfaces:

- ATM
- Basic Rate Interface (BRI)
- CTunnel
- Dialer
- Ethernet
- Fast Ethernet
- Gigabit Ethernet
- High-Speed Serial Interface (HSSI)
- Null
- Packet-over-SONET (POS)
- Serial
- Tunnel
- VLAN

The following interface types can be configured as local interfaces:

- Async

- Bridge Group Virtual Interface (BVI)
- Code division multiple access Internet exchange (CDMA-Ix)
- CTunnel
- Dialer
- Ethernet
- Group-Async
- Loopback
- Multilink
- Multilink Frame Relay (MFR)
- Null
- Serial
- Tunnel
- Virtual host interface (Vif)
- Virtual-PPP
- Virtual-Template
- Virtual-TokenRing

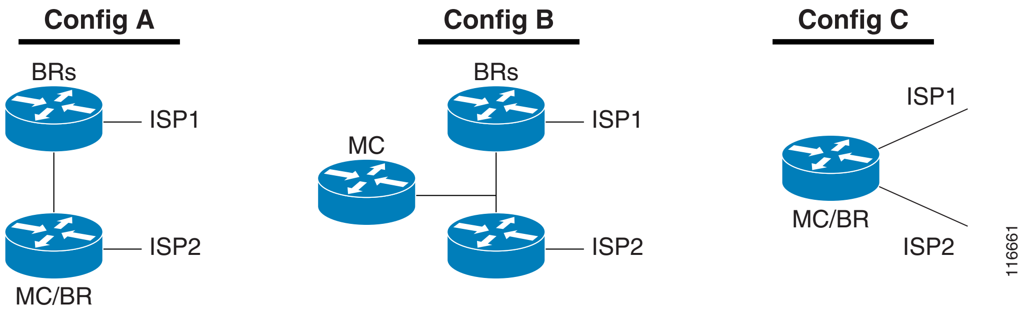
Note A virtual-TokenRing interface can be configured as a local interface. However, Token Ring interfaces are not supported and cannot be configured as external, internal, or local interfaces.

OER Deployment Scenarios

OER can be deployed in an enterprise network, remote office network, or small office home office (SOHO) network using one of the following three configurations shown in Figure 3:

- Configuration A shows a network with two edge routers configured as BRs. The border router that peers with ISP2 is also configured to run a master controller process. This configuration is suitable for a small or medium network with multiple edge routers, each of which provides an exit link to a separate external network.
- Configuration B shows two border routers and a master controller, each running on a separate router. This configuration is suitable for small, medium, and large networks. In this configuration, the master controller process is run on a separate Cisco router. This router performs no routing or forwarding functions, although routing and forwarding functions are not prohibited.
- Configuration C shows a single router that is configured to run a master controller and border router process. This configuration is suitable for a small network with a single router, such as a remote office or home network.

Figure 3 OER Deployment Scenarios



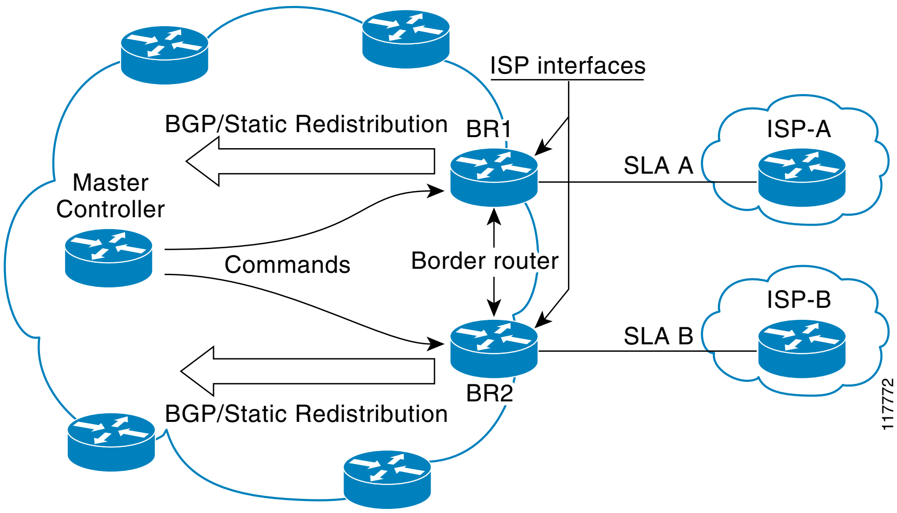
In each deployment scenario, a single master controller is deployed. The master controller does not have to be in the traffic forwarding path but must be reachable by the border routers. A master controller process can be enabled on router that is also configured to run a border router process. The master controller can support up to 10 border routers and up to 20 OER-managed external interfaces. At least one border router process and two external interfaces are required in an OER-managed network.

Note A Cisco router that is configured to run both a master controller and border router process will use more memory than a router that is configured to run only a border router process. This memory impact should be considered when selecting a router for dual operation.

Routing Control Using OER

Figure 4 shows an OER-managed network. The master controller alters IPv4 routing behavior inside of the OER-managed network to optimize traffic class and exit link performance. OER uses a command and response protocol to manage all communication between the border router and the master controller. The border routers are enterprise edge routers. Routing protocol peering or static routing is established between the border routers and internal peers. The border routers advertise a default route to internal peers through BGP peering, static routing, or route redistribution into an IGP. The master controller alters routing behavior in the OER-managed network by sending control commands to the border routers to inject a preferred route into the internal network.

Figure 4 OER Controls Default Routing Behavior Through Peering or Redistribution



When the master controller determines the best exit for a traffic class prefix, it sends a route control command to the border router with the best exit. The border router searches for a parent route for the monitored prefix. The BGP routing table is searched before the static routing table. The parent route can be a default route for the monitored prefix. If a parent route is found that includes the prefix (the parent route prefix may be equivalent or less specific than the original prefix) and points to the desired exit link by either the route to its next hop or by a direct reference to the interface, a preferred route is injected into the internal network from the border router. OER injects the preferred route where the first parent is found. The preferred route can be an injected BGP route or an injected static route. The preferred route is learned by internal peers, which in turn recalculate their routing tables, causing the monitored prefix to be moved to the preferred exit link. The preferred route is advertised only to the internal network, not to external peers.

Border Router Peering with the Internal Network

The master controller alters default routing behavior in the OER-managed network by injecting preferred routes into the routing tables of the border routers. The border routers peer with other routers in the internal network through BGP peering, BGP or static route redistribution into an IGP, or static routing. The border routers advertise the preferred route to internal peers.

The border routers should be close to one another in terms of hops and throughput and should have a consistent view of the network; routing should be configured consistently across all border routers. The master controller verifies that a monitored prefix has a parent route with a valid next hop before it commands the border routers to alter routing. The border router will not inject a route where one does not already exist. This behavior is designed to prevent traffic from being lost because of an invalid next hop.

Note When two or more border routers are deployed in an OER-managed network, the next hop on each border router, as installed in the RIB, cannot be an IP address from the same subnet.

BGP Peering with OER

Standard iBGP peering can be established between the border routers and other internal peers. External BGP (eBGP) peering or a default route is configured to the ISP. In an iBGP network, the local preference attribute is used to set the preference for injected routes. Local preference is a discretionary attribute that is used to apply the degree of preference to a route during BGP best-path selection. This attribute is exchanged only between iBGP peers and is not advertised outside of the OER-managed network or to eBGP peers. The prefix with the highest local preference value is locally advertised as the preferred path to the destination. OER applies a local preference value of 5000 to injected routes by default. A local preference value from 1 to 65535 can be configured.

Note If a local preference value of 5000 or higher has been configured for default BGP routing, you should configure a higher local preference value in OER using the **mode** command in OER master controller configuration mode.

Note In Cisco IOS Release 12.4(6)T and prior releases, the IP address for each eBGP peering session must be reachable from the border router via a connected route. Peering sessions established through loopback interfaces or with the **neighbor ebgp-multihop** command are not supported. In Cisco IOS Release 12.4(9)T and 12.2(33)SRB, the **neighbor ebgp-multihop** command is supported.

BGP Redistribution into an IGP

BGP redistribution can be used if the border routers are configured to run BGP (for ISP peering for example) and the internal peers are configured to run another routing protocol (such as Enhanced Interior Gateway Routing Protocol [EIGRP], Open Shortest Path First [OSPF] or Routing Information Protocol [RIP]). The border routers can advertise a single, default route or full routing tables to the internal network. If you use BGP to redistribute more than a default route into an IGP, we recommend that you use IP prefix-list and route-map statements to limit the number of redistributed prefixes (BGP routing tables can be very large).

Static Routing and Static Route Redistribution into an IGP

Static routing or static route redistribution can be configured in the internal network. OER alters routing for this type of network by injecting temporary static routes. The temporary static route replaces the parent static route. OER will not inject a temporary static route where a parent static route does not exist. OER applies a default tag value of 5000 to identify the injected static route. In a network where only static routing is configured, no redistribution configuration is required. In a network where an IGP is deployed and BGP is not run on the border routers, static routes to border router exit interfaces must be configured, and these static routes must be redistributed into the IGP.

Caution Caution must be applied when redistributing OER static routes into an IGP. The routes injected by OER may be more specific than routes in the IGP, and it will appear as if the OER border router is originating these routes. To avoid routing loops, the redistributed OER static routes should never be advertised over a WAN by an OER border router or any other router. Route filtering and stub network configuration can be used to prevent advertising the OER static routes. If the OER static routes are redistributed to routers terminating the OER external interfaces, routing loops may occur.

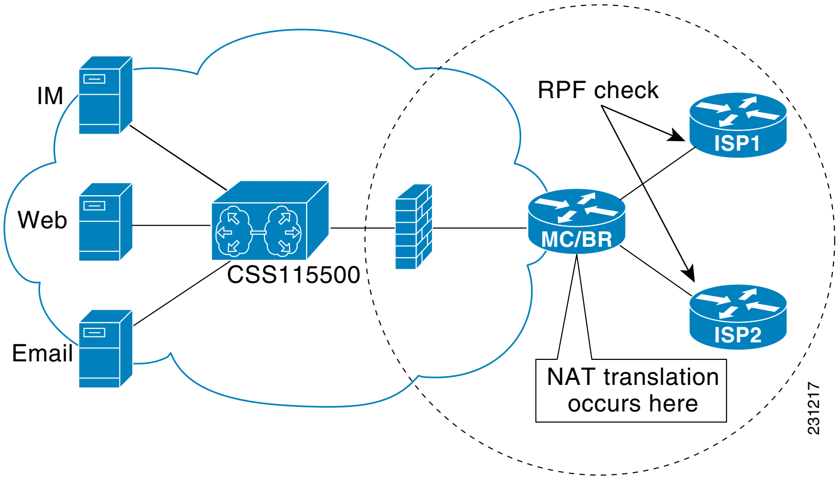
Split Prefixes Injected into the Routing Table

When configured to control a subset of a larger network, the master controller will add an appropriate route or split prefix to the existing routing table, as necessary. A split prefix is a more specific route that is derived from a less specific parent prefix. For example, if a /24 prefix is configured to be optimized, but only a /16 route is installed to the routing table, the master controller will inject a /24 prefix using the attributes of the /16 prefix. Any subset of the less-specific prefix can be derived, including a single host route. Split prefixes are processed only inside the OER-managed network and are not advertised to external networks. If BGP is deployed in the OER-managed network, the master controller will inject a more specific BGP route. If BGP is not deployed, the master controller will inject a more specific temporary static route.

OER and NAT

When Cisco IOS OER and NAT functionality are configured on the same router and OER controls the routing for a traffic class using static routing, some applications may fail to operate due to dropped packets. This dropping of packets behavior is seen when static routing is used to connect to multiple ISPs from the same router, OER uses static routing to control the traffic class routing, and one or more of the ISPs use Unicast Reverse Path Forwarding (Unicast RPF) filtering for security reasons. Packets are dropped at the ingress router performing Unicast RPF because OER changes the route for an outgoing packet for a traffic class from one exit interface to another after the NAT translation from a private IP address to a public IP address is performed. When the packet is transmitted, Unicast RPF filtering at the ingress router (for example, an ISP router) will show a different source IP address from the source IP address pool assigned by NAT, and the packet is dropped. For example, Figure 5 shows how OER works with NAT.

Figure 5 OER with NAT



The NAT translation occurs at the router that is connected to the internal network, and this router can be a border router or a combined master controller and border router. If OER changes routes to optimize traffic class performance and to perform load balancing, traffic from the border router in Figure 5 that was routed through the interface to ISP1 may be rerouted through the interface to ISP2 after the traffic performance is measured and policy thresholds are applied. The RPF check occurs at the ISP routers and any packets that are now routed through ISP2 will fail the RPF check at the ingress router for ISP2 because the IP address of the source interface has changed.

The solution involves a minimal configuration change with a new keyword, **oer**, that has been added to the **ip nat inside source** command. When the **oer** keyword is configured, new NAT translations are given the source IP address of the interface that OER has selected for the packet and OER forces existing flows to be routed through the interface for which the NAT translation was created. For example, OER is configured to manage traffic on a border router with two interfaces, InterfaceA to ISP1 and InterfaceB to ISP2 in Figure 5. OER is first configured to control a traffic class representing Web traffic and the NAT translation for this traffic already exists with the source IP address in the packets set to InterfaceA. OER measures the traffic performance and determines that InterfaceB is currently the best exit for traffic flows, but OER does not change the existing flow. When OER is then configured to learn and measure a traffic class representing e-mail traffic, and the e-mail traffic starts, the NAT translation is done for InterfaceB. The OER static routing NAT solution is a single box solution and configurations with interfaces on multiple routers using NAT and managed by OER are not supported. Network configurations using NAT and devices such as PIX firewalls that do not run Cisco IOS software are not supported.

For details about configuring the OER static routing NAT solution, see the "Configuring OER to Control Traffic with Static Routing in Networks Using NAT" task.

OER Application Interface

In Cisco IOS Release 12.4(15)T support for an OER application interface was introduced. The OER application interface defines the mode of communication and messaging between applications and the network for the purpose of optimizing the traffic associated with the applications. A provider is defined as an entity outside the network in which the router configured as an OER master controller exists, for example, an ISP, or a branch office of the same company. The provider has one or more host devices running one or more applications that use the OER application interface to communicate with an OER master controller. A provider must be registered with an OER master controller before an application on a host device can interface with OER. Host devices in the provider network running an application that communicates with OER must also be configured at an OER master controller with an IP address and key chain password.

After registration, a host device in the provider network can initiate a session with an OER master controller. When a provider application initiates a session with an OER master controller, a session identifier (ID) number is allocated to the session. After a session is established, the application can send a request for reports containing performance numbers for traffic classes, dynamically create policies to influence the existing traffic classes, or specify new traffic class criteria.

The application interface can be used by Cisco partners to develop applications. An example of application developed by a partner is OER Manager by Fluke Networks. OER Manager is a complete graphical-user interface (GUI) interface for the Optimized Edge Routing technology. It provides detailed reporting on traffic class performance and OER behavior as well as easy-to-use configuration of OER traffic classes and policies. For more details about OER Manager, go to http://www.flukenetworks.com/pfr.

The OER application interface permits a maximum of five concurrent sessions, and keepalives are used to check that the session between the host application device and the OER master controller is still active. If the session is dropped, all policies created in the session are dropped. An application may negotiate an ability for the session to persist in the case of a temporary outage.

Application Interface Priority

The OER application interface has three main levels of priority to help resolve conflicts with requests coming from providers, host devices, and policies. In Table 1 the three priority levels are shown with the scope of the priority, whether the priority level can be configured on the master controller, the range and default values, if applicable.

When multiple providers are registered with OER, an optional priority value can be specified to give OER the ability to order requests coming in from multiple providers. Host devices in a provider network can also be assigned a priority. The lower the priority value, the higher the priority. If you configure a priority, each provider must be assigned a different priority number. If you try to assign the same priority number to two different providers, an error message is displayed on the console. Host devices must also be configured with different priority numbers if a priority is configured. If a priority has not been configured for the provider or host device, the priority is set to the default value of 65535, which is the lowest priority.

Table 1 Application Interface Priority Level Table

Priority Name	Scope	Mandatory In Application Interface Message	Configure on MC	Default Value	Range
Provider priority	Network wide	No	Yes	65535	1 to 65535
Host priority	Provider Level	No	Yes	65535	1 to 65535
Policy	Host Level	Yes	No	N/A	1 to 65535

The application administrator assigns a priority to all applications. This priority is conveyed to the Network in terms of a policy priority. The lower the application priority number, the higher the priority of the application. Policy priority is handled using the policy sequence number. A policy sequence number—see Table 2—is a 64 bit number calculated by placing provider priority in bytes 1 and 2, host priority in bytes 3 and 4, policy priority in bytes 5 and 6 and Session ID in bytes 7 and 8. The policy sequence number is calculated by the OER master controller. An example policy sequence number is 18446744069421203465, representing a provider priority value of 65535, a host priority of 65535, a policy priority of 101, and a session ID of 9.

Use the **show oer master policy** command to view the policy sequence number. The lower the sequence number, the higher the priority for the policy.

Table 2 Formulation of a Policy Sequence Number

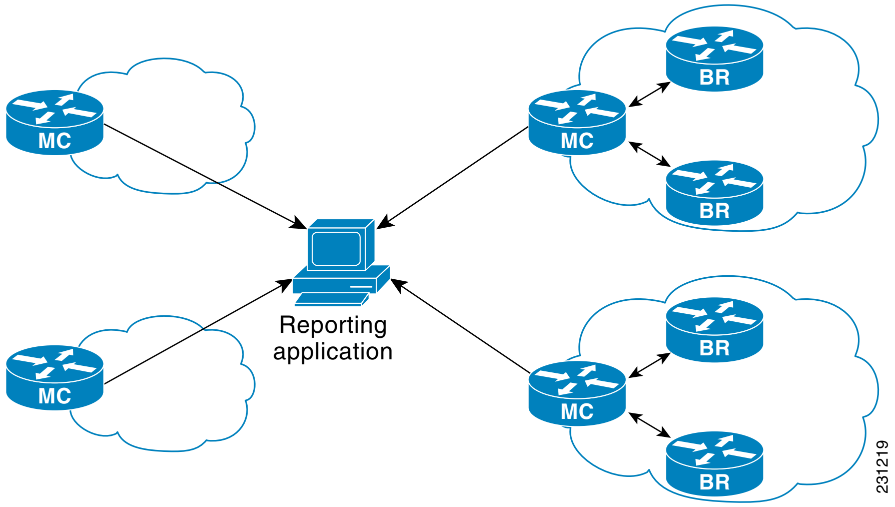
Bits 0-7	**Bits 8-15**	**Bits 16-23**	**Bits 24-31**
Provider Priority		Host Priority	
Bits 32-39	**Bits 40-48**	**Bits 49-56**	**Bits 57-64**
Policy Priority		Session ID	

In the situation where an application tries to create two policies with same policy priority; the second policy creation attempt will fail.

OER Application Interface Reporting Deployment

An application communicating through an OER application interface can request performance reports from OER and use the report information to create graphs and charts of the information. Figure 6 shows a diagram of an example reporting model. In this example, the topology contains multiple sites using OER within the site. Each site has a master controller but the company wants to review reports about activities in each site such as overall inter-site traffic activity, voice and video traffic activity, and data center access reports. An OER application interface solution is implemented with a reporting application—see Figure 6—that resides in a central location. The reporting application is registered at each OER master controller and the application initiates a session with each master controller and requests traffic class performance information. The master controller at each site exports information to the application, which consolidates the information and displays graphs and charts. Reports can be requested at specified intervals to keep the information on the reporting application updated.

Figure 6 OER Application Interface Reporting Model



At each site the master controller can monitor provider activity. Several Cisco IOS command-line interface (CLI) commands allow you to view provider information including details about dynamic policies created by the application. Reporting can also be implemented for a single site.

In summary, the OER application interface provides an automated method for networks to be aware of applications and provides application-aware performance routing.

OER Logging and Reporting

Cisco IOS OER supports standard syslog functions. The notice level of syslog is enabled by default. System logging is enabled and configured in Cisco IOS software under global configuration mode. The **logging** command in OER master controller or OER border router configuration mode is used only to enable or disable system logging under OER. OER system logging supports the following message types:

- Error Messages—These messages indicate OER operational failures and communication problems that can impact normal OER operation.
- Debug Messages—These messages are used to monitor detailed OER operations to diagnose operational or software problems.
- Notification Messages—These messages indicate that OER is performing a normal operation.
- Warning Messages—These messages indicate that OER is functioning properly but an event outside of OER may be impacting normal OER operation.

To modify system, terminal, destination, and other system global logging parameters, use the logging commands in global configuration mode. For more information about global system logging configuration, see to the “Troubleshooting, Logging, and Fault Management” section of the *Cisco IOS Network Management Configuration Guide*.

How to Set Up OER Network Components

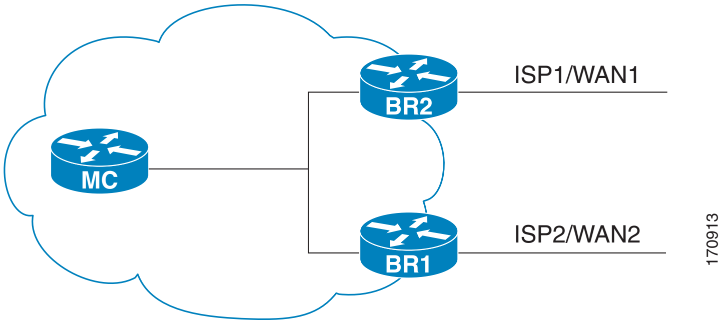
To set up an OER-managed network you must configure routing protocol peering or redistribution between border routers and peer routers in order for OER to control routing. Perform the first two tasks to set up the OER master controller and OER border routers. After performing these required tasks, the other tasks are optional and depend on the existing routing configuration in your network. For example, if only static routing is configured in your network, no optional configuration tasks are necessary for initial OER configuration.

- Setting Up the OER Master Controller, page 15
- Setting Up an OER Border Router, page 21
- Configuring an Interim Border Router, page 25
- Configuring OER to Control Traffic with Static Routing in Networks Using NAT, page 28
- Configuring iBGP Peering on the Border Routers Managed by OER, page 31
- Redistributing BGP Routes into an IGP in an OER-Managed Network, page 33
- Redistributing Static Routes into an IGP in an OER-Managed Network, page 36
- Redistributing Static Routes into EIGRP in an OER-Managed Network, page 39
- Registering an Application Interface Provider and Configuring Host Devices, page 42
- Displaying Information about Application Interface Provider Activity, page 44

Setting Up the OER Master Controller

Perform this task to set up the OER master controller to manage an OER-managed network. This task must be performed on the router designated as the OER master controller. For an example network configuration of a master router and two border routers, see Figure 7. Communication is first established between the master controller and the border routers with key-chain authentication being configured to protect the communication session between the master controller and the border routers. Internal and external border router interfaces are also specified.

Figure 7 Master Controller and Border Router Diagram



Key Chain Authentication for OER

Communication between the master controller and the border router is protected by key-chain authentication. The authentication key must be configured on both the master controller and the border router before communication can be established. The key-chain configuration is defined in global configuration mode on both the master controller and the border router before key-chain authentication is enabled for master controller-to-border router communication. For more information about key management in Cisco IOS software, see the "Managing Authentication Keys" section of the "Configuring IP Routing Protocol-Independent Features" chapter in the *Cisco IOS IP Routing Protocols Configuration Guide*.

Master Controller Process Disablement

To disable a master controller and completely remove the process configuration from the running configuration, use the **no oer master** command in global configuration mode.

To temporarily disable a master controller, use the **shutdown** command in OER master controller configuration mode. Entering the **shutdown** command stops an active master controller process but does not remove any configuration parameters. The **shutdown** command is displayed in the running configuration file when enabled.

Manual Port Configuration

Communication between the master controller and border router is automatically carried over port 3949 when connectivity is established. Port 3949 is registered with the Internet Assigned Numbers Authority (IANA) for OER communication. Support for port 3949 was introduced in Cisco IOS Release 12.3(11)T and 12.2(33)SRB. Manual port number configuration is required only if you are running Cisco IOS Release 12.3(8)T or if you need to configure OER communication to use a dynamic port number.

Prerequisites

Interfaces must be defined and reachable by the master controller and the border router before an OER-managed network can be configured.

Note Token Ring interfaces are not supported by OER and cannot be configured as OER-managed interfaces. It may be possible to load a Token Ring interface configuration under certain conditions. However, the Token Ring interface will not become active, and the border router will not function if the Token Ring interface is the only external interface on the border router.

Tip We recommend that the master controller be physically close to the border routers to minimize communication response time in OER-managed networks. If traffic is to be routed between border routers, the border routers also should be physically close each other to minimize the number of hops.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **key chain** *name-of-chain*
4. **key** *key-id*
5. **key-string** *text*
6. **exit**

7. **exit**
8. Repeat Step 3 through Step 7 with appropriate changes to configure key chain authentication for each border router.
9. **oer master**
10. **port** *port-number*
11. **logging**
12. **border** *ip-address* [**key-chain** *key-chain-name*]
13. **interface** *type number* **external**
14. **exit**
15. **interface** *type number* **internal**
16. **exit**
17. Repeat Step 12 through Step 16 with appropriate changes to establish communication with each border router.
18. **keepalive** *timer*
19. **end**
20. **show running-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`key chain`** *`name-of-chain`*
	Example:
	`Router(config)# key chain border1_OER`
		Enables key-chain authentication and enters key-chain configuration mode.
		• Key-chain authentication protects the communication session between the master controller and the border router. The key ID and key string must match in order for communication to be established.
		• In this example, a key chain is created for use with border router 1.

Step 4	**`key`** *`key-id`*
	Example:
	`Router(config-keychain)# key 1`
		Identifies an authentication key on a key chain.
		• The key ID must match the key ID configured on the border router.

	Command or Action	Purpose

Step 5	**`key-string`** *`text`*
	Example:
	`Router(config-keychain-key)# key-string b1`
		Specifies the authentication string for the key and enters key-chain key configuration mode.
		- The authentication string must match the authentication string configured on the border router.
		- Any encryption level can be configured.
		- In this example, a key string is created for use with border router 1.

Step 6	**`exit`**
	Example:
	`Router(config-keychain-key)# exit`
		Exits key-chain key configuration mode and returns to key-chain configuration mode.

Step 7	**`exit`**
	Example:
	`Router(config-keychain)# exit`
		Exits key-chain configuration mode and returns to global configuration mode.

Step 8	Repeat Step 3 through Step 7 with appropriate changes to configure key chain authentication for each border router.	—

Step 9	**`oer master`**
	Example:
	`Router(config)# oer master`
		Enters OER master controller configuration mode to configure a router as a master controller.
		- A master controller and border router process can be enabled on the same router (for example, in a network that has a single router with two exit links to different service providers).
		Note Only the syntax used in this context is displayed. For more details, see the *Cisco IOS Optimized Edge Routing Command Reference*.

Step 10	**`port`** *`port-number`*
	Example:
	`Router(config-oer-mc)# port 65534`
		(Optional) Configures a dynamic port for communication between the master controller and border router.
		- Communication cannot be established until the same port number has been configured on both the master controller and the border router.
		Note Manual port number configuration is required to establish OER communication only when running Cisco IOS Release 12.3(8)T.

Step 11	**`logging`**
	Example:
	`Router(config-oer-mc)# logging`
		Enables syslog messages for a master controller or border router process.
		- The notice level of syslog messages is enabled by default.

	Command or Action	Purpose
Step 12	**border** *ip-address* [**key-chain** *key-chain-name*] **Example:** `Router(config-oer-mc)# border 10.1.1.2 key-chain border1_OER`	Enters OER-managed border router configuration mode to establish communication with a border router. • An IP address is configured to identify the border router. • At least one border router must be specified to create an OER-managed network. A maximum of ten border routers can be controlled by a single master controller. • The value for the *key-chain-name* argument must match the key-chain name configured in Step 3. **Note** The **key-chain** keyword and *key-chain-name* argument must be entered when a border router is initially configured. However, this keyword is optional when reconfiguring an existing border router.
Step 13	**interface** *type number* **external** **Example:** `Router(config-oer-mc-br)# interface Ethernet 1/0 external`	Configures a border router interface as an OER-managed external interface. • External interfaces are used to forward traffic and for active monitoring. • A minimum of two external border router interfaces are required in an OER-managed network. At least one external interface must be configured on each border router. A maximum of 20 external interfaces can be controlled by single master controller. **Tip** Configuring an interface as an OER-managed external interface on a router enters OER border exit interface configuration mode. In this mode, you can configure maximum link utilization or cost-based optimization for the interface. **Note** Entering the **interface** command without the **external** or **internal** keyword places the router in global configuration mode and not OER border exit configuration mode. The **no** form of this command should be applied carefully so that active interfaces are not removed from the router configuration.
Step 14	**exit** **Example:** `Router(config-oer-mc-br-if)# exit`	Exits OER-managed border exit interface configuration mode and returns to OER-managed border router configuration mode.

	Command or Action	Purpose
Step 15	**interface** *type number* **internal** **Example:** `Router(config-oer-mc-br)# interface Ethernet 0/0 internal`	Configures a border router interface as an OER controlled internal interface. • Internal interfaces are used for passive monitoring only. Internal interfaces do not forward traffic. • At least one internal interface must be configured on each border router. **Note** Support to configure a VLAN interface as an internal interface was introduced in Cisco IOS Release 12.3(14)T and 12.2(33)SRB.
Step 16	**exit** **Example:** `Router(config-oer-mc-br)# exit`	Exits OER-managed border router configuration mode and returns to OER master controller configuration mode.
Step 17	Repeat Step 12 through Step 16 with appropriate changes to establish communication with each border router.	—
Step 18	**keepalive** *timer* **Example:** `Router(config-oer-mc)# keepalive 10`	(Optional) Configures the length of time that an OER master controller will maintain connectivity with an OER border router after no keepalive packets have been received. • The example sets the keepalive timer to 10 seconds. The default keepalive timer is 60 seconds.

Step 19	**end**
	Example:
	`Router(config-oer-mc-learn)# end`
		Exits OER Top Talker and Top Delay learning configuration mode and returns to privileged EXEC mode.

Step 20	**show running-config** **Example:** `Router# show running-config`	(Optional) Displays the running configuration to verify the configuration entered in this task.

Examples

The following partial output shows the section of the running configuration file that contains the OER master controller configuration from this task. A second border router was also identified.

```
Router# show running-config

!
key chain border1_OER
 key 1
  key-string b1
key chain border2_OER
 key 1
  key-string b2
oer master
 port 65534
 keepalive 10
 logging
 !
 border 10.1.1.2 key-chain border1_OER
  interface Ethernet0/0 internal
  interface Ethernet1/0 external
 !
 border 10.1.1.3 key-chain border2_OER
  interface Ethernet0/0 internal
  interface Ethernet1/0 external
.
.
.
```

Setting Up an OER Border Router

Perform this task to set up an OER border router. This task must be performed at each border router in your OER-managed network. For an example network configuration of a master router and two border routers, see Figure 7. Communication is first established between the border router and the master controller with key-chain authentication being configured to protect the communication session between the border router and the master controller. A local interface is configured as the source for communication with the master controller, and external interfaces are configured as OER-managed exit links.

Interface Configuration in an OER-Managed Network

- Each border router must have at least one external interface that is either used to connect to an ISP or is used as an external WAN link. A minimum of two external interfaces are required in an OER-managed network.
- Each border router must have at least one internal interface. Internal interfaces are used for only passive performance monitoring with NetFlow. Internal interfaces are not used to forward traffic.
- Each border router must have at least one local interface. Local interfaces are used only for master controller and border router communication. A single interface must be configured as a local interface on each border router.

Tip If a master controller and border router process is enabled on the same router, a loopback interface should be configured as the local interface.

Disabling a Border Router Process

To disable a border router and completely remove the process configuration from the running configuration, use the **no oer border** command in global configuration mode.

To temporarily disable a border router process, use the **shutdown** command in OER border router configuration mode. Entering the **shutdown** command stops an active border router process but does not remove any configuration parameters. The **shutdown** command is displayed in the running configuration file when enabled.

Prerequisites

Perform the task "Setting Up the OER Master Controller" section on page 15 to set up the master controller and define the interfaces and establish communication with the border routers.

Tip We recommend that the border routers be physically close to one another to minimize the number of hops. The master controller also should be physically close to the border routers to minimize communication response time in OER-managed networks.

Restrictions

- Internet exchange points where a border router can communicate with several service providers over the same broadcast media are not supported.
- When two or more border routers are deployed in an OER-managed network, the next hop to an external network on each border router, as installed in the RIB, cannot be an IP address from the same subnet.
- In Cisco IOS Release 12.2(33)SXH support for using a Cisco Catalyst 6500 series switch as an OER border router was introduced. Only border router functionality is included in the Cisco IOS Release 12.2(33)SXH images; no master controller configuration is available. The master controller that communicates with the Cisco Catalyst 6500 series switch being used as a border router must be a router running Cisco IOS Release 12.4(6)T or a later release.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **key chain** *name-of-chain*
4. **key** *key-id*
5. **key-string** *text*
6. **exit**
7. **exit**
8. **oer border**
9. **port** *port-number*
10. **local** *type number*
11. **master** *ip-address* **key-chain** *key-chain-name*
12. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`key chain`** *`name-of-chain`*
	Example:
	`Router(config)# key chain border1_OER`
		Enables key-chain authentication and enters key-chain configuration mode.
		• Key-chain authentication protects the communication session between both the master controller and the border router. The key ID and key string must match in order for communication to be established.

Step 4	**`key`** *`key-id`*
	Example:
	`Router(config-keychain)# key 1`
		Identifies an authentication key on a key chain and enters key-chain key configuration mode.
		• The key ID must match the key ID configured on the master controller.

Step 5	**`key-string`** *`text`*
	Example:
	`Router(config-keychain-key)# key-string b1`
		Specifies the authentication string for the key.
		• The authentication string must match the authentication string configured on the master controller.
		• Any level of encryption can be configured.

Step 6	**`exit`**
	Example:
	`Router(config-keychain-key)# exit`
		Exits key-chain key configuration mode and returns to key-chain configuration mode.

Step 7	**`exit`**
	Example:
	`Router(config-keychain)# exit`
		Exits key-chain configuration mode and returns to global configuration mode.

Step 8	**`oer border`**
	Example:
	`Router(config)# oer border`
		Enters OER border router configuration mode to configure a router as a border router.
		• The border router must be in the forwarding path and contain at least one external and internal interface.
		Note Only the syntax used in this context is displayed. For more details, see the *Cisco IOS Optimized Edge Routing Command Reference*.

	Command or Action	Purpose
Step 9	**port** *port-number* **Example:** `Router(config-oer-br)# port 65534`	(Optional) Configures a dynamic port for communication between an OER master controller and border router. • Communication cannot be established until the same port number has been configured on both the border router and the master controller. **Note** Manual port number configuration is required to establish OER communication only when running Cisco IOS Release 12.3(8)T.
Step 10	**local** *type number* **Example:** `Router(config-oer-br)# local Ethernet 0/0`	Identifies a local interface on an OER border router as the source for communication with an OER master controller. • A local interface must be defined. **Tip** A loopback should be configured when a single router is configured to run both a master controller and border router process.
Step 11	**master** *ip-address* **key-chain** *key-chain-name* **Example:** `Router(config-oer-br)# master 10.1.1.1 key-chain border1_OER`	Enters OER-managed border router configuration mode to establish communication with a master controller. • An IP address is used to identify the master controller. • The value for the *key-chain-name* argument must match the key-chain name configured in Step 3.

Step 12	**end**
	Example:
	`Router(config-oer-br)# end`
		Exits OER Top Talker and Top Delay learning configuration mode and returns to privileged EXEC mode.

What to Do Next

If the design of your network requires any border routers to be separated by more than one hop, a Cisco router between the border routers can be configured as an interim border router. Support for an interim border router was introduced in Cisco IOS Release 12.4(2)T, 12.2(33)SRB, and later releases. To configure an interim border router. proceed to the "Configuring an Interim Border Router" task.

If your network is configured to use only static routing, no additional configuration is required. The OER-managed network should be operational, as long as valid static routes that point to external interfaces on the border routers are configured. You can proceed to the "Where to Go Next" section at the end of this document for information about further OER customization.

Otherwise, routing protocol peering or static redistribution must be configured between the border routers and other routers in the OER-managed network.

The master controller implements policy changes by altering IP routing behavior in the OER-managed network. If iBGP peering is enabled on the border routers, the master controller will inject iBGP routes into routing tables on the border routers. To configure iBGP peering on the border routers managed by OER, proceed to the "Configuring iBGP Peering on the Border Routers Managed by OER" task.

If BGP is configured on the border routers and another IGP is deployed in the internal network, proceed to the "Redistributing BGP Routes into an IGP in an OER-Managed Network" task for more information about configuring redistribution from BGP into the IGP.

If BGP is not configured in the internal network, then static routes to the border exits must be configured and the static routes must be redistributed into the IGP. For more information, see the "Redistributing Static Routes into an IGP in an OER-Managed Network" task.

If you need to configure static redistribution into EIGRP, see the "Redistributing Static Routes into EIGRP in an OER-Managed Network" task for more information.

Configuring an Interim Border Router

In Cisco IOS Release 12.4(2)T and 12.2(33)SRB, support for a border router that is more than one hop away from another border router was introduced. In releases prior to Cisco IOS Release 12.4(2)T, the border routers must be one hop away from each other. However, if the design of your network requires the border routers to be separated by more than one hop, a Cisco router between the border routers can be configured as an interim border router. The interim border routers act as transit routers between the border routers in your network. The master controller discovers the paths between interim and standard border routers and policy routes traffic through the appropriate external interface on a standard border router.

Perform this task on the master controller to configure an interim border router. The configuration of an interim border router is similar to standard border router configuration. The physical configuration on the interim border router is the same as on a standard border router. The difference is in the configuration on the master controller. Only a single internal interface is configured. No external interface configuration is required.

Prerequisites

- Perform the task "Setting Up an OER Border Router" section on page 21 to set up the Cisco router that will act as an interim border router.
- This task requires the master controller and border routers to be running Cisco IOS Release 12.4(2)T, 12.2(33)SRB, or later release.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **key chain** *name-of-chain*
4. **key** *key-id*
5. **key-string** *text*
6. **exit**
7. exit
8. **oer master**
9. **border** *ip-address* [**key-chain** *key-chain-name*]
10. **interface** *type number* **internal**
11. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`key chain`** *`name-of-chain`*
	Example:
	`Router(config)# key chain OER`
		Enables key-chain authentication and enters key chain configuration mode.
		• Key-chain authentication protects the communication session between the master controller and the border router. The key ID and key string must match in order for communication to be established.

Step 4	**`key`** *`key-id`*
	Example:
	`Router(config-keychain)# key 1`
		Identifies an authentication key on a key chain.
		• The key ID must match the key ID configured on the border router.

Step 5	**`key-string`** *`text`*
	Example:
	`Router(config-keychain-key)# key-string CISCO`
		Specifies the authentication string for the key.
		• The authentication string must match the authentication string configured on the border router.
		• Any encryption level can be configured.

Step 6	**`exit`**
	Example:
	`Router(config-keychain-key)# exit`
		Exits key chain key configuration mode and enters key chain configuration mode.

Step 7	**`exit`**
	Example:
	`Router(config-keychain)# exit`
		Exits key chain configuration mode and returns to global configuration mode.

Step 8	**`oer master`**
	Example:
	`Router(config)# oer master`
		Enters OER master controller configuration mode to configure a router as a master controller.
		• In this example, the master controller is configured
		Note Only the syntax used in this context is displayed. For more details, see the *Cisco IOS Optimized Edge Routing Command Reference*.

	Command or Action	Purpose
Step 9	**border** *ip-address* [**key-chain** *key-chain-name*] **Example:** `Router(config-oer-mc)# border 10.100.1.1 key-chain OER`	Enters OER-managed border router configuration mode to establish communication with a border router. • An IP address is configured to identify the border router. • At least one border router must be specified to create an OER-managed network. A maximum of 10 border routers can be controlled by a single master controller. • The value for the *key-chain-name* argument must match the key-chain name configured in Step 3. **Note** The **key-chain** keyword and argument must be entered when a border router is initially configured. However, this keyword is optional when reconfiguring an existing border router.
Step 10	**interface** *type number* **internal** **Example:** `Router(config-oer-mc-br)# interface Ethernet 0/1 internal`	Configures a border router interface as an OER controlled internal interface. • Internal interfaces are used for passive monitoring only. Internal interfaces do not forward traffic. • At least one internal interface must be configured on each border router.

Step 11	**end**
	Example:
	`Router(config-oer-mc-br)# end`
		Exits OER-managed border router configuration mode and returns to privileged EXEC mode.

What to Do Next

If your network is configured to use only static routing without NAT, no additional configuration is required. The OER-managed network should be operational, as long as valid static routes that point to external interfaces on the border routers are configured. You can proceed to the "Where to Go Next" section at the end of this document for information about further OER customization.

If your network is configured to use NAT and static routing is used by OER to control traffic classes, proceed to the "Configuring OER to Control Traffic with Static Routing in Networks Using NAT" task.

Otherwise, routing protocol peering or static redistribution must be configured between the border routers and other routers in the OER-managed network.

The master controller implements policy changes by altering IP routing behavior in the OER-managed network. If iBGP peering is enabled on the border routers, the master controller will inject iBGP routes into routing tables on the border routers. To configure iBGP peering on the border routers managed by OER, proceed to the "Configuring iBGP Peering on the Border Routers Managed by OER" task.

If BGP is configured on the border routers and another IGP is deployed in the internal network, proceed to the "Redistributing BGP Routes into an IGP in an OER-Managed Network" task for more information about configuring redistribution from BGP into the IGP.

If BGP is not configured in the internal network, then static routes to the border exits must be configured and the static routes must be redistributed into the IGP. For more information, see the "Redistributing Static Routes into an IGP in an OER-Managed Network" task.

If you need to configure static redistribution into EIGRP, see the "Redistributing Static Routes into EIGRP in an OER-Managed Network" task for more information.

Configuring OER to Control Traffic with Static Routing in Networks Using NAT

Perform this task to allow OER to control traffic with static routing in a network using NAT. This task allows OER to optimize traffic classes while permitting your internal users access to the internet.

When Cisco IOS OER and NAT functionality are configured on the same router and OER controls the routing for a traffic class using static routing, some applications may fail to operate due to dropped packets. This dropping of packets behavior is seen when static routing is used to connect to multiple ISPs from the same router, OER uses static routing to control the traffic class routing, and one or more of the ISPs use Unicast Reverse Path Forwarding (Unicast RPF) filtering for security reasons.

In this task, the **oer** keyword is used with the **ip nat inside source** command. When the **oer** keyword is configured, new NAT translations are given the source IP address of the interface that OER has selected for the packet and OER forces existing flows to be routed through the interface where the NAT translation was created. This task uses a single IP address but an IP address pool can also be configured. For a configuration example using an IP address pool, see "Configuring OER to Control Traffic with Static Routing in Networks Using NAT: Example" section on page 48.

Note The OER static routing NAT solution is a single box solution and configurations with interfaces on multiple routers using NAT and managed by OER are not supported.

For more details about configuring NAT, see the "Configuring NAT for IP Address Conservation" chapter of the *Cisco IOS IP Addressing Services Configuration Guide*.

NAT

NAT enables private IP internetworks that use nonregistered IP addresses to connect to the Internet. NAT operates on a router, usually connecting two networks together, and translates the private (not globally unique) address in the internal network into legal addresses before packets are forwarded onto another network. NAT can be configured to advertise only one address for the entire network to the outside world. This ability provides additional security, effectively hiding the entire internal network behind that one address.

NAT is also used at the Enterprise edge to allow internal users access to the Internet and to allow Internet access to internal devices such as mail servers.

Inside Global Addresses Overloading

You can conserve addresses in the inside global address pool by allowing the router to use one global address for many local addresses. When this overloading is configured, the router maintains enough information from higher-level protocols (for example, TCP or UDP port numbers) to translate the global address back to the correct local address. When multiple local addresses map to one global address, the TCP or UDP port numbers of each inside host distinguish between the local addresses.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **access-list** *access-list-number* {**permit** | **deny**} *ip-address mask*
4. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
5. **match ip address** {**access-list** *access-list-number* | **prefix-list** *prefix-list-name*}
6. **match interface** *interface-type interface-number* [...*interface-type interface-number*]
7. **exit**
8. Repeat Step 4 through Step 7 for more route map configurations, as required.
9. **ip nat inside source** {**list** {*access-list-number* | *access-list-name*} | **route-map** *map-name*} {**interface** *type number* | **pool** *name*} [**mapping-id** *map-id* | **overload** | **reversible** | **vrf** *vrf-name*] [**oer**]
10. **interface** *type number*
11. **ip address** *ip-address mask*
12. **ip nat inside**
13. **exit**
14. **interface** *type number*
15. **ip address** *ip-address mask*
16. **ip nat outside**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`access-list`** *`access-list-number`* `{`**`permit`** `|` **`deny`**`}` *`ip-address`* `mask`
	Example:
	`Router(config)# access-list 1 permit 10.1.0.0 0.0.255.255`
		Defines a standard access list permitting the IP addresses that are to be translated.
		• The access list must permit only those addresses that are to be translated. (Remember that there is an implicit "deny all" at the end of each access list.) An access list that is too permissive can lead to unpredictable results.

Step 4	**`route-map`** *`map-tag`* `[`**`permit`** `|` **`deny`**`]` `[`*`sequence-number`*`]`
	Example:
	`Router(config)# route-map isp-1 permit 10`
		Enters route-map configuration mode to configure a route map.
		• The example creates a route map named BGP.

	Command or Action	Purpose

Step 5	**match ip address** {**access-list** *access-list-name* | **prefix-list** *prefix-list-name*}
	Example:
	`Router(config-route-map)# match ip address access-list 1`
		Creates an access list or prefix list match clause entry in a route map to identify traffic to be translated by NAT.
		• The example references the access list created in Step 3 that specifies the 10.1.0.0 0.0.255.255. prefix as match criteria.

Step 6	**match interface** *interface-type interface-number* [...*interface-type interface-number*] **Example:** `Router(config-route-map)# match interface serial 1/0`	Creates a match clause in a route map to distribute any routes that match out one of the interfaces specified. • The example creates a match clause to distribute routes that pass the match clause in Step 5 through serial interface 1/0.
Step 7	**exit** **Example:** `Router(config-route-map)# exit`	Exits route-map configuration mode and returns to global configuration mode.
Step 8	Repeat Step 4 through Step 7 for more route map configurations, as required.	—
Step 9	**ip nat inside source** {**list** {*access-list-number* | *access-list-name*} | **route-map** *map-name*} {**interface** *type number* | **pool** *name*} [**mapping-id** *map-id* | **overload** | **reversible** | **vrf** *vrf-name*] [**oer**] **Example:** `Router(config)# ip nat inside source interface FastEthernet1/0 overload oer`	Establishes dynamic source translation with overloading, specifying the interface. • Use the **interface** keyword and type and number arguments to specify an interface. • Use the **oer** keyword to allow OER to operate with NAT and control traffic class routing using static routing.
Step 10	**interface** *type number* **Example:** `Router(config)# interface FastEthernet1/0`	Specifies an interface and enters interface configuration mode.
Step 11	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.114.11.8 255.255.255.0`	Sets a primary IP address for the interface.
Step 12	**ip nat inside** **Example:** `Router(config-if)# ip nat inside`	Marks the interface as connected to the inside.
Step 13	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to configuration mode.

	Command or Action	Purpose
Step 14	**interface** *type number* **Example:** `Router(config)# interface ethernet 0`	Specifies a different interface and returns to interface configuration mode.
Step 15	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 172.17.233.208 255.255.255.0`	Sets a primary IP address for the interface.
Step 16	**ip nat outside** **Example:** `Router(config-if)# ip nat outside`	Marks the interface as connected to the outside.

What to Do Next

Routing protocol peering or static redistribution must be configured between the border routers and other routers in the OER-managed network.

The master controller implements policy changes by altering IP routing behavior in the OER-managed network. If iBGP peering is enabled on the border routers, the master controller will inject iBGP routes into routing tables on the border routers. To configure iBGP peering on the border routers managed by OER, proceed to the "Configuring iBGP Peering on the Border Routers Managed by OER" task.

If BGP is configured on the border routers and another IGP is deployed in the internal network, proceed to the "Redistributing BGP Routes into an IGP in an OER-Managed Network" task for more information about configuring redistribution from BGP into the IGP.

If BGP is not configured in the internal network, then static routes to the border exits must be configured and the static routes must be redistributed into the IGP. For more information, see the "Redistributing Static Routes into an IGP in an OER-Managed Network" task.

If you need to configure static redistribution into EIGRP, see the "Redistributing Static Routes into EIGRP in an OER-Managed Network" task for more information.

Configuring iBGP Peering on the Border Routers Managed by OER

Perform this task at each border router to configure iBGP peering on the border routers managed by OER. The master controller implements policy changes by altering IP routing behavior in the OER-managed network. If iBGP peering is enabled on the border routers, the master controller will inject iBGP routes into routing tables on the border routers. The border routers advertise the preferred route through standard iBGP peering.

The local preference attribute is used to set the preference for injected BGP prefixes. If a local preference value of 5000 or higher has been configured for default BGP routing, you should configure a higher value in OER. Default local preference and static tag values are configurable with the **mode** command in OER master controller configuration mode.

All OER injected routes remain local to an autonomous system. The no-export community is automatically applied to injected routes to ensure that they are not advertised to external networks. Before injecting a route, the master controller verifies that a parent route with a valid next hop exists. This behavior is designed to prevent traffic from being lost.

Prerequisites

Routing protocol peering must be established in your network and consistently applied to the border routers; the border routers should have a consistent view of the network.

Restrictions

In Cisco IOS Release 12.4(6)T and prior releases, the IP address for each eBGP peering session must be reachable from the border router via a connected route. Peering sessions established through loopback interfaces or with the **neighbor ebgp-multihop** command are not supported. In Cisco IOS Release 12.4(9)T and 12.2(33)SRB, the **neighbor ebgp-multihop** command is supported.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vpnv4** [**unicast**]
5. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
6. **neighbor** {*ip-address* | *peer-group-name*} **activate**
7. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	`enable`
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	`configure terminal`
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	`router bgp` *autonomous-system-number* **Example:** `Router(config)# router bgp 65534`	Enters router configuration mode to create or configure a BGP routing process.

	Command or Action	Purpose
Step 4	**address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vpnv4** [**unicast**] **Example:** `Router(config-router)# address-family ipv4 unicast`	Enters address-family configuration mode to configure a BGP address family session. • The example creates an IPv4 unicast address family session.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number* **Example:** `Router(config-router)# neighbor 10.100.1.3 remote-as 65534`	Establishes BGP peering with the specified neighbor or border router.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Router(config-router)# neighbor 10.100.1.3 activate`	Enables the exchange of routing information under an address family.

Step 7	**end**
	Example:
	`Router(config-router)# end`
		Exits router configuration mode and returns to privileged EXEC mode.

What to Do Next

If BGP is configured on the border routers and another IGP is deployed in the internal network, proceed to the "Redistributing BGP Routes into an IGP in an OER-Managed Network" task for more information about configuring redistribution from BGP into the IGP.

If BGP is not configured in the internal network, then static routes to the border exits must be configured and the static routes must be redistributed into the IGP. For more information, see the "Redistributing Static Routes into an IGP in an OER-Managed Network" task.

If you need to configure static redistribution into EIGRP, see the "Redistributing Static Routes into EIGRP in an OER-Managed Network" task for more information.

Redistributing BGP Routes into an IGP in an OER-Managed Network

This task explains how to redistribute BGP routes into an IGP in an OER-managed network. Some of the examples in the "Detailed Steps" section of this task show redistribution into OSPF, but EIGRP, IS-IS, or RIP could also be used in this configuration.

When redistributing BGP routes into any IGP, be sure to use the **ip prefix-list** and **route-map** command statements to limit the number of prefixes. Redistributing full BGP routing tables into an IGP can have a detrimental effect on IGP network operation.

Prerequisites

IGP peering, static routing, and static route redistribution must be applied consistently throughout the OER-managed network; the border routers should have a consistent view of the network.

Restrictions

When two or more border routers are deployed in an OER-managed network, the next hop to an external network on each border router, as installed in the RIB, cannot be an IP address from the same subnet.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network***/***length* **|** **permit** *network***/***length*} [**ge** *ge-value*] [**le** *le-value*]
4. **ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network***/***length* **|** **permit** *network***/***length*} [**ge** *ge-value*] [**le** *le-value*]
5. **route-map** *map-tag* [**permit** **|** **deny**] [*sequence-number*]
6. **match ip address prefix-list** *prefix-list-name*
7. **exit**
8. **router bgp** *autonomous-system-number*
9. **bgp redistribute-internal**
10. **exit**
11. **router** {**eigrp** *autonomous-system-number* **|** **is-is** [*area-tag*] **|** **ospf** *process-id* **|** **rip**}
12. **redistribute static** [**metric** *metric-value*] [**route-map** *map-tag*] [**subnets**]
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network***/***length* | **permit** *network***/***length*} [**ge** *ge-value*] [**le** *le-value*] **Example:** `Router(config)# ip prefix-list PREFIXES seq 5 permit 10.200.2.0/24`	Defines the prefix range to redistribute into the IGP. • Any prefix length can be specified. • The first longest match is processed in the IP prefix list. • This example creates a prefix list named PREFIXES and the entry permits the 10.200.2.0/24 subnet.

	Command or Action	Purpose
Step 4	**ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network***/***length* | **permit** *network***/***length*} [**ge** *ge-value*] [**le** *le-value*] **Example:** `Router(config)# ip prefix-list PREFIXES seq 10 deny 0.0.0.0/0`	Defines additional prefix list entries. • Any prefix length can be specified. • The first longest match is processed in the IP prefix list. • This example prefix list entry denies all other prefixes.
Step 5	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Router(config)# route-map BGP permit 10`	Enters route-map configuration mode to configure a route map. • The example creates a route map named BGP.
Step 6	**match ip address prefix-list** *prefix-list-name* **Example:** `Router(config-route-map)# match ip address prefix-list PREFIXES`	Creates a prefix list match clause entry in a route map to redistribute BGP prefixes. • The example references the prefix list named PREFIXES as match criteria.
Step 7	**exit** **Example:** `Router(config-route-map)# exit`	Exits route-map configuration mode and returns to global configuration mode.
Step 8	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 65534`	Enters router configuration mode to configure a BGP routing process.
Step 9	**bgp redistribute-internal** **Example:** `Router(config-router)# bgp redistribute-internal`	Enables BGP redistribution into an IGP.
Step 10	**exit** **Example:** `Router(config-router)# exit`	Exits router configuration mode and returns to global configuration mode.

Step 11	**router** {**eigrp** *autonomous-system-number* | **is-is** [*area-tag*] | **ospf** *process-id* | **rip**}
	Example:
	`Router(config)# router ospf 1`
		Enters router configuration mode and creates a routing process.
		• The example creates an OSPF routing process.

	Command or Action	Purpose

Step 12	`redistribute static [metric metric-value] [route-map map-tag] [subnets]`
	Example:
	`Router(config-router)# redistribute static route-map BGP subnets`
		Redistributes static routes into the specified protocol.
		• The example configures the IGP to accept the redistributed BGP routes that pass through the route map.
		• In OSPF, the **subnets** keyword must be entered if you redistribute anything less than a major network
		Note Only the syntax used in this context is displayed. For more details, see the *Cisco IOS IP Routing Protocols Command Reference*.

Step 13	`end`
	Example:
	`Router(config-router)# end`
		Exits router configuration mode and returns to privileged EXEC mode.

What to Do Next

The master controller implements policy changes by altering default routing behavior in the OER-managed network. If iBGP peering is enabled on the border routers, the master controller will inject iBGP routes into routing tables on the border routers.

If BGP is not configured in the internal network, then static routes to the border exits must be configured and the static routes must be redistributed into the IGP. For more information, see the "Redistributing Static Routes into an IGP in an OER-Managed Network" task.

If you need to configure static redistribution into EIGRP, see the "Redistributing Static Routes into EIGRP in an OER-Managed Network" task for more information.

Redistributing Static Routes into an IGP in an OER-Managed Network

This task shows how to redistribute static routes into an IGP in an OER-managed network. This task should be performed on the border routers.

OER applies a default tag value of 5000 to injected temporary static routes. The static route is filtered through a route map and then redistributed into the IGP. If you use the tag value of 5000 for another routing function, you should use a different tag value for that function, or you can change the default static tag values by configuring the **mode** command in OER master controller configuration mode.

Before injecting a route, the master controller verifies that a parent route with a valid next hop exists. This behavior is designed to prevent traffic from being lost.

If static routing is configured in your network and no IGP is deployed, OER will inject temporary static routes as necessary. No redistribution or other specific network configuration is required.

The following IGPs are supported; EIGRP, OSPF, Intermediate System-to-Intermediate System (IS-IS), and RIP.

Caution Caution must be applied when redistributing OER static routes into an IGP. The routes injected by OER may be more specific than routes in the IGP, and it will appear as if the OER border router is originating these routes. To avoid routing loops, the redistributed OER static routes should never be advertised over a WAN by an OER border router or any other router. Route filtering and stub network configuration can be used to prevent advertising the OER static routes. If the OER static routes are redistributed to routers terminating the OER external interfaces, routing loops may occur.

Note OER supports static route redistribution into EIGRP; however, it is configured differently. Proceed to the "Redistributing Static Routes into EIGRP in an OER-Managed Network" task for more information.

Prerequisites

IGP peering, static routing, and static route redistribution must be applied consistently throughout the OER-managed network; the border routers should have a consistent view of the network.

Restrictions

When two or more border routers are deployed in an OER-managed network, the next hop to an external network on each border router, as installed in the RIB, cannot be an IP address from the same subnet as the next hop on the other border router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [*distance*] [*name*] [**permanent**] [**tag** *tag*]
4. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
5. **match tag** *tag-value* [*...tag-value*]
6. **set metric** *metric-value*
7. **exit**
8. **router** {**is-is** *area-tag* | **ospf** *process-id* | **rip**}
9. **redistribute static** [**metric** *metric-value*] [**route-map** *map-tag*]
10. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [*distance*] [*name*] [**permanent**] [**tag** *tag*] **Example:** `Router(config)# ip route 0.0.0.0 0.0.0.0 Ethernet 0`	Configures a static route. • A static route must be configured for each external interface. The static route is configured only on the border routers. The static route must include any prefixes that need to be optimized.
Step 4	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Router(config)# route-map STATIC permit 10`	Enters route-map configuration mode and creates a route map. • The example creates a route map named STATIC.
Step 5	**match tag** *tag-value* [*...tag-value*] **Example:** `Router(config-route-map)# match tag 5000`	Redistribute routes in the routing table that match the specified tag value. • 5000 must be configured for this tag value unless you have configured a different value with the **mode** command.
Step 6	**set metric** *metric-value* **Example:** `Router(config-route-map)# set metric -10`	Sets the metric value for prefixes that pass through the route map. • A metric value that is less than 1 must be configured in order for the OER injected static route to be preferred by default routing. • The example set the metric value for the OER injected routes to -10.
Step 7	**exit** **Example:** `Router(config-route-map)# exit`	Exits route-map configuration mode and returns to global configuration mode.

Step 8	**router** {**is-is** *area-tag* | **ospf** *process-id* | **rip**}
	Example:
	`Router(config)# router rip`
		Enters router configuration mode and creates a routing process for the specified routing protocol.

Step 9	**redistribute static** [**metric** *metric-value*] [**route-map** *map-tag*] **Example:** `Router(config-router)# redistribute static route-map STATIC`	Redistributes static routes into the specified protocol. • The example configures the IGP to redistribute static routes injected from the REDISTRIBUTE_STATIC route map. **Note** In OSPF, the **subnets** keyword must be entered if you redistribute anything less than a major network.

Step 10	**end**
	Example:
	`Router(config-router)# end`
		Exits router configuration mode and returns to privileged EXEC mode.

What to Do Next

If you need to configure static redistribution into EIGRP, see the "Redistributing Static Routes into EIGRP in an OER-Managed Network" task for more information.

Redistributing Static Routes into EIGRP in an OER-Managed Network

This task explains how to redistribute static routes into EIGRP. For EIGRP configurations, a tag is applied to the static route and the tag is then filtered through a route map. Two route map sequences are configured in this task. A route map named BLUE is configured to permit both configured static routes and OER static routes, and BLUE is the route map used to redistribute both types of static routes into EIGRP. A route map named RED is configured to permit only the configured static routes and implicitly deny the OER static routes. A distribute list uses the RED route map to filter outbound advertisements on the Ethernet 0 and Ethernet 1 egress interfaces. By denying the OER static route outbound advertisements, routing loops can be avoided.

OER applies a default tag value of 5000 to injected temporary static routes. The static route is filtered through a route map and then redistributed into the IGP.

Before injecting the temporary static route, the master controller verifies that a parent static route with a valid next hop exists. This behavior is designed to prevent traffic from being lost.

Caution Caution must be applied when redistributing OER static routes into an IGP. The routes injected by OER may be more specific than routes in the IGP, and it will appear as if the OER border router is originating these routes. To avoid routing loops, the redistributed OER static routes should never be advertised over a WAN by an OER border router or any other router. Route filtering and stub network configuration can be used to prevent advertising the OER static routes. If the OER static routes are redistributed to routers terminating the OER external interfaces, routing loops may occur.

Prerequisites

IGP peering, static routing, and static route redistribution must be applied consistently throughout the OER-managed network; the border routers should have a consistent view of the network.

Restrictions

When two or more border routers are deployed in an OER-managed network, the next hop, as installed in the RIB, to an external network on each border router cannot be an IP address from the same subnet.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [*distance*] [*name*] [**permanent**] [**tag** *tag*]
4. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
5. **match tag** *tag-value* [*...tag-value*]
6. **match tag** *tag-value* [*...tag-value*]

7. **exit**
8. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
9. **match tag** *tag-value* [*...tag-value*]
10. **exit**
11. **router eigrp** *autonomous-system-number*
12. **no auto-summary**
13. **network** *ip-address* [*wildcard-mask*]
14. **redistribute static** [**metric** *metric-value*] [**route-map** *map-tag*]
15. **distribute-list** {*acl-number* | *acl-name* | *prefix-list-name*} **out** [*interface-name* | *routing-process* | *autonomous-system-number*]
16. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [*distance*] [*name*] [**permanent**] [**tag** *tag*] **Example:** `Router(config)# ip route 0.0.0.0 0.0.0.0 Ethernet 0 tag 10`	Configures a static route. • A static route must be configured for each external interface. The static route is configured only on the border routers. The static route must include any prefixes that need to be optimized. • Under EIGRP, a tag is applied to the static route. The tag is then filtered through a route map.
Step 4	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Router(config)# route-map BLUE permit 10`	Enters route-map configuration mode and creates a route map. • A route map named BLUE is configured.
Step 5	**match tag** *tag-value* [*...tag-value*] **Example:** `Router(config-route-map)# match tag 5000`	Redistributes additional routes in the routing table that match the specified tag value. • This example matches the default OER tag value applied to injected temporary static routes.
Step 6	**match tag** *tag-value* [*...tag-value*] **Example:** `Router(config-route-map)# match tag 10`	Redistributes routes in the routing table that match the specified tag value. • This example matches the configured static route tag.

	Command or Action	Purpose
Step 7	**exit** **Example:** Router(config-route-map)# exit	Exits route-map configuration mode and returns to global configuration mode.
Step 8	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** Router(config)# route-map RED permit 10	Enters route-map configuration mode and creates a route map. • A route map named RED is configured.
Step 9	**match tag** *tag-value* [*...tag-value*] **Example:** Router(config-route-map)# match tag 10	Redistributes routes in the routing table that match the specified tag value. • This example matches the configured static route tag.
Step 10	**exit** **Example:** Router(config-route-map)# exit	Exits route-map configuration mode and returns to global configuration mode. • By exiting route map configuration mode with no deny statements, an implicit deny is in effect for the OER static routes.
Step 11	**router eigrp** *autonomous-system-number* **Example:** Router(config)# router eigrp 1	Enters router configuration mode and creates an EIGRP routing process.
Step 12	**no auto-summary** **Example:** Router(config-router)# no auto-summary	Disables automatic summarization under the EIGRP routing process.
Step 13	**network** *ip-address* [*wildcard-mask*] **Example:** Router(config-router)# network 192.168.0.0 0.0.255.255	Specifies a network for an EIGRP routing process. • The network state must cover any interfaces and prefixes that have to be optimized for the internal network.
Step 14	**redistribute static** [**metric** *metric-value*] [**route-map** *map-tag*] **Example:** Router(config-router)# redistribute static route-map BLUE	Redistributes static routes into the specified protocol. • The example configures redistribution of static routes that are filtered through the route map named BLUE, into EIGRP. • Both configured static and OER static routes are redistributed.

	Command or Action	Purpose
Step 15	**distribute-list** {*acl-number* | *acl-name* | *prefix-list-name*} **out** [*interface-name* | *routing-process* | *autonomous-system-number*] **Example:** `Router(config-router)# distribute-list RED out Ethernet 0`	Applies a distribute list to filter outbound advertisements. • The distribute list must be applied to egress interfaces. • Using the route map named RED, the OER static routes are filtered out of outbound advertisements on Ethernet interface 0.

Step 16	**end**
	Example:
	`Router(config-router)# end`
		Exits router configuration mode and returns to privileged EXEC mode.

Registering an Application Interface Provider and Configuring Host Devices

Perform this task at a master controller to register an application interface provider with the master controller and to configure host devices. In Cisco IOS Release 12.4(15)T the OER application interface was introduced. The OER application interface defines the mode of communication and messaging between applications and the network for the purpose of optimizing the traffic associated with the applications. A provider must be registered with an OER master controller before the application can interface with OER.

Multiple providers can be registered and multiple host devices can be configured under each provider, but a host device cannot be configured under multiple providers. The OER application interface has a maximum number of five concurrent sessions. After the provider is registered using this task, an application running on a host device can initiate a session with the master controller.

To view information about providers and any default policies created by applications using the OER application interface, see the "Displaying Information about Application Interface Provider Activity" section on page 44. For more details about the OER application interface, see "OER Application Interface" section on page 12.

Prerequisites

The master controller and border routers must be running Cisco IOS Release 12.4(15)T, or later release.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **oer master**
4. **api provider** *provider-id* [**priority** *value*]
5. **host-address** *ip-address* [**key-chain** *key-chain-name*] [**priority** *value*]
6. Repeat Step 5 to configure additional host devices as required.
7. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`oer master`**
	Example:
	`Router(config)# oer master`
		Enters OER master controller configuration mode to configure a router as a master controller.
		• A master controller and border router process can be enabled on the same router (for example, in a network that has a single router with two exit links to different service providers).
		Note Only the syntax used in this context is displayed. For more details, see the *Cisco IOS Optimized Edge Routing Command Reference*.

Step 4	**api provider** *provider-id* [**priority** *value*] **Example:** `Router(config-oer-mc)# api provider 1 priority 3000`	Registers a provider with an OER master controller and enters OER master controller application interface provider configuration mode. • Use the **priority** keyword to assign a priority for this provider when there are multiple providers. The lower the number, the higher the priority. The default priority is 65535, the lowest priority. • In this example, the provider is assigned an ID of 1 and a priority of 3000.
Step 5	**host-address** *ip-address* [**key-chain** *key-chain-name*] [**priority** *value*] **Example:** `Router(config-oer-mc-api-provider)# host-address 10.1.2.2 key-chain OER_HOST1`	Configures information about a host device used by a provider to communicate with an OER master controller. • Use the **priority** keyword to assign a priority for this host device when there are multiple host devices. The lower the number, the higher the priority. The default priority is 65535, the lowest priority. • In this example, the host IP address of 10.1.2.2 is configured, the key chain password is set to OER_HOST1, and the priority is not configured and will be set to the default value of 65535.
Step 6	Repeat Step 5 to configure additional host devices as required.	—

Step 7	**`end`**
	Example:
	`Router(config-router)# end`
		Exits OER master controller application interface provider configuration mode and returns to privileged EXEC mode.

Troubleshooting Tips

Use the **debug oer api** command on the master controller to troubleshoot issues with registering a provider or configuring a host device. Use the **detailed** keyword with caution in a production network.

Displaying Information about Application Interface Provider Activity

Perform this task on a master controller to display information about providers and any default policies created by applications using the OER application interface. In Cisco IOS Release 12.4(15)T the OER application interface was introduced. The OER application interface defines the mode of communication and messaging between applications and the network for the purpose of optimizing the traffic associated with the applications. This task can be used after a provider is registered with an OER master controller using the Registering an Application Interface Provider and Configuring Host Devices task and an application on a host device initiates a session. The **show** commands can be entered in any order.

Prerequisites

- The master controller and border routers must be running Cisco IOS Release 12.4(15)T, or later release.
- Perform the Registering an Application Interface Provider and Configuring Host Devices task and run an application from a host device using the OER application interface.

SUMMARY STEPS

1. **enable**
2. **show oer api provider** [**detail**]
3. **show oer master policy** [*sequence-number* | *policy-name* | **default** | **dynamic**]
4. **show oer master prefix** [**detail** | **inside** [**detail**] | **learned** [**delay** | **inside** | **throughput**] | *prefix* [**detail** | **policy** | **report** | **traceroute** [*exit-id* | *border-address* | **current**] [**now**]]]

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

```
Router> enable
```

Step 2 **show oer api provider** [**detail**]

This command is used to display provider and host information including the ID of each configured provider, the priority of the provider and the host (if configured), and the IP addresses of each configured host device.

```
Router# show oer api provider detail

API Version: Major 2, Minor 0
  Provider id 1001, priority 65535
   Host ip 10.3.3.3, priority 65535
    Session id 9, Version Major 2, Minor 0
    Num pfx created 2, Num policies created 2
    Last active connection time (sec) 00:00:01
    Policy ids : 101, 102,
   Host ip 10.3.3.4, priority 65535
    Session id 10, Version Major 2, Minor 0
    Num pfx created 1, Num policies created 1
    Last active connection time (sec) 00:00:03
    Policy ids : 103,
  Provider id 2001, priority 65535
   Host ip 172.19.198.57, priority 65535
    Session id 11, Version Major 2, Minor 0
    Num pfx created 0, Num policies created 0
    All Prefix report enabled
    All exit report enabled
```

Step 3 **show oer master policy** [*sequence-number* | policy-name | **default** | **dynamic**]

This command is used to display policy information. The following example uses the **dynamic** keyword to display the policies dynamically created by provider applications. Note that the first two dynamic policies were generated by the same host device at 10.3.3.3 and in the same session ID of 9, but the third section is for a different host device at 10.3.3.4.

```
Router# show oer master policy dynamic

Dynamic Policies:

  proxy id 10.3.3.3
  sequence no. 18446744069421203465, provider id 1001, provider priority 65535
    host priority 65535, policy priority 101, Session id 9
  backoff 90 90 90
  delay relative 50
  holddown 90
  periodic 0
  probe frequency 56
  mode route control
  mode monitor both
  mode select-exit good
  loss relative 10
  jitter threshold 20
  mos threshold 3.60 percent 30
  unreachable relative 50
  next-hop not set
  forwarding interface not set
  resolve delay priority 11 variance 20
  resolve utilization priority 12 variance 20

proxy id 10.3.3.3
  sequence no. 18446744069421269001, provider id 1001, provider priority 65535
    host priority 65535, policy priority 102, Session id 9
  backoff 90 90 90
  delay relative 50
  holddown 90
  periodic 0
  probe frequency 56
  mode route control
  mode monitor both
  mode select-exit good
  loss relative 10
  jitter threshold 20
  mos threshold 3.60 percent 30
  unreachable relative 50
  next-hop not set
  forwarding interface not set
  resolve delay priority 11 variance 20
  resolve utilization priority 12 variance 20

  proxy id 10.3.3.4
  sequence no. 18446744069421334538, provider id 1001, provider priority 65535
    host priority 65535, policy priority 103, Session id 10
  backoff 90 90 90
  delay relative 50
  holddown 90
  periodic 0
  probe frequency 56
  mode route control
  mode monitor both
  mode select-exit good
  loss relative 10
  jitter threshold 20
  mos threshold 3.60 percent 30
  unreachable relative 50
  next-hop not set
  forwarding interface not set
  resolve delay priority 11 variance 20
  resolve utilization priority 12 variance 20
```

Step 4 **show oer master prefix** [**detail** | **inside** [**detail**] | **learned** [**delay** | **inside** | **throughput**] | *prefix* [**detail** | **policy** | **report** | **traceroute** [*exit-id* | *border-address* | **current**] [**now**]]]

This command is used to display the status of monitored prefixes. Using the **report** keyword, the following example shows prefix statistics including information about provider report requests for the 10.1.1.0 prefix:.

```
Router# show oer master prefix 10.1.1.0/24 report

Prefix Performance Report Request
   Created by: Provider 1001, Host 10.3.3.3, Session 9
   Last report sent 3 minutes ago, context 589855, frequency 4 min

Prefix Performance Report Request
   Created by: Provider 1001, Host 10.3.3.4, Session 10
   Last report sent 1 minutes ago, context 655372, frequency 3 min

OER Prefix Statistics:
 Pas - Passive, Act - Active, S - Short term, L - Long term, Dly - Delay (ms),
 P - Percentage below threshold, Jit - Jitter (ms),
 MOS - Mean Opinion Score
 Los - Packet Loss (packets-per-million), Un - Unreachable (flows-per-million),
 E - Egress, I - Ingress, Bw - Bandwidth (kbps), N - Not applicable
 U - unknown, * - uncontrolled, + - control more specific, @ - active probe all
 # - Prefix monitor mode is Special, & - Blackholed Prefix
 % - Force Next-Hop, ^ - Prefix is denied

Prefix                  State     Time Curr BR         CurrI/F         Protocol
                      PasSDly  PasLDly   PasSUn   PasLUn  PasSLos  PasLLos
                      ActSDly  ActLDly   ActSUn   ActLUn      EBw      IBw
                      ActSJit  ActPMOS  ActSLos  ActLLos
--------------------------------------------------------------------------------
10.1.1.0/24             INPOLICY        0 10.3.3.3        Et4/3           BGP
                               N        N        N        N        N        N
                             138      145        0        0        N        N
                               N        N
```

Configuration Examples for Setting Up OER Network Components

This section contains the following examples:

- Configuring the OER Master Controller: Example, page 47
- Configuring an OER Border Router: Example, page 48
- Configuring an Interim Border Router: Example, page 48
- Configuring OER to Control Traffic with Static Routing in Networks Using NAT: Example, page 48
- Configuring iBGP Peering on the Border Routers Managed by OER: Example, page 49
- Redistributing BGP Routes into an IGP in an OER-Managed Network: Example, page 50
- Redistributing Static Routes into an IGP in an OER-Managed Network: Example, page 50
- Redistributing Static Routes into EIGRP in an OER-Managed Network: Example, page 51
- OER Master Controller and Two Border Routers Deployment: Example, page 51
- OER Master Controller and Border Router Process Deployed on a Single Router with a Second Border Router: Example, page 54
- OER Master Controller and Border Router Deployed on a Single Router: Example, page 55
- Registering an Application Interface Provider and Configuring Host Devices: Example, page 56

Configuring the OER Master Controller: Example

The following configuration example, starting in global configuration mode, shows the minimum configuration required to configure a master controller process to manage the internal network. A key-chain configuration named OER is defined in global configuration mode.

```
Router(config)# key chain OER
Router(config-keychain)# key 1
Router(config-keychain-key)# key-string KEYSTRING2
Router(config-keychain-key)# end
```

The master controller is configured to communicate with the 10.100.1.1 and 10.200.2.2 border routers. The keepalive interval is set to 10 seconds. Route control mode is enabled. Internal and external OER-controlled border router interfaces are defined.

```
Router(config)# oer master
Router(config-oer-mc)# keepalive 10
Router(config-oer-mc)# logging
Router(config-oer-mc)# border 10.100.1.1 key-chain OER
Router(config-oer-mc-br)# interface Ethernet 0/0 external
Router(config-oer-mc-br)# interface Ethernet 0/1 internal
Router(config-oer-mc-br)# exit
Router(config-oer-mc)# border 10.200.2.2 key-chain OER
Router(config-oer-mc-br)# interface Ethernet 0/0 external
Router(config-oer-mc-br)# interface Ethernet 0/1 internal
Router(config-oer-mc)# exit
```

Configuring an OER Border Router: Example

The following configuration example, starting in global configuration mode, shows the minimum required configuration to enable a border router. The key-chain configuration is defined in global configuration mode.

```
Router(config)# key chain OER
Router(config-keychain)# key 1
Router(config-keychain-key)# key-string KEYSTRING2
Router(config-keychain-key)# end
```

The key-chain OER is applied to protect communication. An interface is identified to the master controller as the local interface (source) for OER communication.

```
Router(config)# oer border
Router(config-oer-br)# local Ethernet 0/1
Router(config-oer-br)# master 192.168.1.1 key-chain OER
Router(config-oer-br)# end
```

Configuring an Interim Border Router: Example

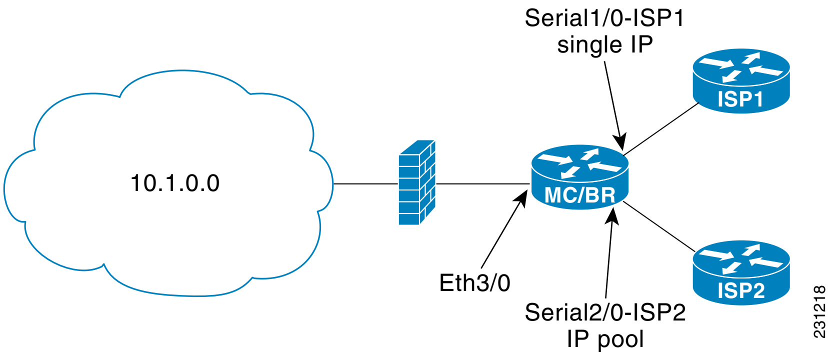
The following configuration example configures an interim border router on a master controller:

```
Router(config)# key chain OER
Router(config-keychain)# key 1
Router(config-keychain-key)# key-string keystring3
Router(config-keychain-key)# exit
Router(config-keychain)# exit
Router(config)# oer master
Router(config-oer-mc)# border 10.100.1.1 key-chain OER
Router(config-oer-mc-br)# interface Ethernet 0/1 internal
Router(config-oer-mc-br)# end
```

Configuring OER to Control Traffic with Static Routing in Networks Using NAT: Example

The following configuration example configures a master controller to allow OER to control traffic with static routing in a network using NAT. This example shows how to use a pool of IP addresses for the NAT translation.

Figure 8 OER and NAT Network Diagram



In Figure 8 there is a combined master controller and border router that is connected to the Internet through two different ISPs. The configuration below allows OER to optimize traffic classes while permitting the internal users access to the internet. In this example the traffic classes to be translated using NAT are specified using an access list and a route map. The use of a pool of IP addresses for NAT translation is then configured and the **oer** keyword is added to the **ip nat inside source** command to configure OER to keep existing traffic classes flowing through the interface that is the source address that was translated by NAT. New NAT translations can be given the IP address of the interface that OER has selected for the packet.

Note The OER static routing NAT solution is a single box solution and configurations with interfaces on multiple routers using NAT and managed by OER are not supported.

```
Router(config)# access-list 1 permit 10.1.0.0 0.0.255.255
Router(config)# route-map isp-2 permit 10BGP permit 10
Router(config-route-map)# match ip address access-list 1
Router(config-route-map)# match interface serial 2/0
Router(config-route-map)# exit
Router(config)# ip nat pool ISP2 209.165.201.1 209.165.201.30 prefix-length 27
Router(config)# ip nat inside source route-map isp-2 pool ISP2 oer
Router(config)# interface FastEthernet 3/0
Router(config-if)# ip address 10.1.11.8 255.255.255.0
Router(config-if)# ip nat inside
Router(config-if)# exit
Router(config)# interface serial 1/0
Router(config-if)# ip address 192.168.3.1 255.255.255.0
Router(config-if)# ip nat outside
Router(config-if)# exit
Router(config)# interface serial 2/0
Router(config-if)# ip address 172.17.233.208 255.255.255.0
Router(config-if)# ip nat outside
Router(config-if)# end
```

For more details about configuring NAT, see the "Configuring NAT for IP Address Conservation" chapter of the *Cisco IOS IP Addressing Services Configuration Guide*.

Configuring iBGP Peering on the Border Routers Managed by OER: Example

The following example, starting in global configuration mode, shows how to establish peering between two routers in autonomous system 65534 and to configure standard community exchange:

Border Router Configuration

```
Router(config)# router bgp 65534
Router(config-router)# neighbor 10.100.1.3 remote-as 65534
Router(config-router)# address-family ipv4
Router(config-router-af)# neighbor 10.100.1.3 activate
Router(config-router-af)# neighbor 10.100.1.3 send-community standard
```

Internal Border Peer Configuration

```
Router(config)# router bgp 65534
Router(config-router)# neighbor 10.100.1.2 remote-as 65534
Router(config-router)# address-family ipv4
Router(config-router-af)# neighbor 10.100.1.2 activate
Router(config-router-af)# neighbor 10.100.1.2 send-community standard
```

Redistributing BGP Routes into an IGP in an OER-Managed Network: Example

The following example, starting in global configuration mode, shows how to configure BGP to OSPF redistribution from the border router. Although this example shows redistribution into OSPF, EIGRP, IS-IS, or RIP could also be used in this configuration.

Note When redistributing BGP routes into any IGP, be sure to use the **ip prefix-list** and **route-map** command statements to limit the number of prefixes. Redistributing full BGP routing tables into an IGP can have a detrimental effect on IGP network operation.

Border Router Configuration

```
Router(config)# ip prefix-list PREFIXES seq 5 permit 10.200.2.0/24
Router(config)# ip prefix-list PREFIXES seq 10 deny 0.0.0.0/0
Router(config)# !
Router(config)# route-map BGP permit 10
Router(config-route-map)# match ip address prefix-list PREFIXES
Router(config-route-map)# exit
Router(config)# router bgp 65534
Router(config-router)# bgp redistribute-internal
```

IGP Peer Configuration

```
Router(config)# router ospf 1
Router(config-router)# redistribute bgp 65534 route-map BGP subnets
```

Redistributing Static Routes into an IGP in an OER-Managed Network: Example

The following example, starting in global configuration mode, shows how to configure static redistribution to allow the master controller to influence routing in an internal network that is running RIP. The **match tag** command is used to match OER-injected temporary static routes. The **set metric** command is used to set the preference of the injected static route.

Border Router Configuration

```
Router(config)# ip route 0.0.0.0 0.0.0.0 Ethernet 0
Router(config)# ip route 0.0.0.0 0.0.0.0 Ethernet 1
Router(config)# route-map STATIC permit 10
Router(config-route-map)# match tag 5000
Router(config-route-map)# set metric -10
Router(config-route-map)# exit
Router(config)# router rip
Router(config-router)# network 192.168.0.0
Router(config-router)# network 172.16.0.0
Router(config-router)# redistribute static route-map STATIC
```

Internal Border Peer Configuration

```
Router(config)# route rip
Router(config-router)# network 192.168.0.0
Router(config-router)# network 172.16.0.0
```

Redistributing Static Routes into EIGRP in an OER-Managed Network: Example

The following example, starting in global configuration mode, shows how to configure static redistribution to allow the master controller to influence routing in an internal network that is running EIGRP. Two route map sequences are configured in this example. A route map named BLUE is configured to permit both configured static routes and OER static routes, and BLUE is the route map used to redistribute both types of static routes into EIGRP. A route map named RED is configured to permit only the configured static routes and implicitly deny the OER static routes. A distribute list uses the RED route map to filter outbound advertisements on the Ethernet 0 and Ethernet 1 egress interfaces. By denying the OER static route outbound advertisements, routing loops can be avoided.

Border Router Configuration

```
Router(config)# ip route 0.0.0.0 0.0.0.0 Ethernet 0 tag 10
Router(config)# ip route 0.0.0.0 0.0.0.0 Ethernet 1 tag 10
Router(config)# route-map BLUE permit 10
Router(config-route-map)# match tag 5000
Router(config-route-map)# match tag 10
Router(config-route-map)# exit
Router(config)# route-map RED permit 20
Router(config-route-map)# match tag 10
Router(config-route-map)# exit
Router(config)# route eigrp 1
Router(config-router)# no auto-summary
Router(config-router)# redistribute static route-map BLUE
Router(config-router)# network 10.0.0.0
Router(config-router)# network 172.16.0.0
Router(config-router)# network 192.168.0.0
Router(config-router)# distribute-list route-map RED out Ethernet 0
Router(config-router)# distribute-list route-map RED out Ethernet 1
```

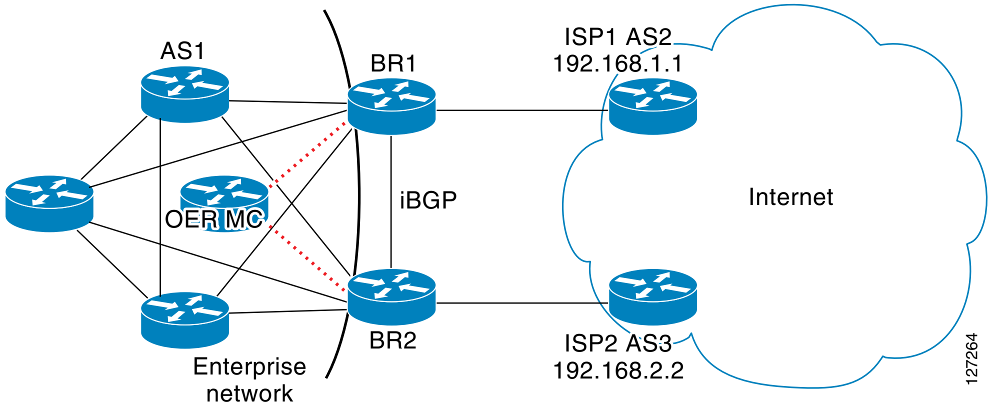
Internal Border Peer Configuration

```
Router(config)# route eigrp 1
Router(config-router)# no auto-summary
Router(config-router)# network 10.0.0.0
Router(config-router)# network 172.16.0.0
Router(config-router)# network 192.168.0.0
Router(config-router)# end
```

OER Master Controller and Two Border Routers Deployment: Example

Figure 9 shows an OER-managed network with two border router processes and a master controller process deployed separately on Cisco routers.

Figure 9 Master Controller Deployed with Two Border Routers



The master controller performs no routing functions. BGP is deployed on the border routers and internal peers in the OER-managed network. Each border router has an eBGP peering session with a different ISP. The eBGP peers (ISP border routers) are reachable through connected routes. Injected prefixes are advertised in the internal network through standard iBGP peering.

OER MC Configuration

The following example, starting in global configuration mode, shows the master controller configuration.

```
Router(config)# key chain OER
Router(config-keychain)# key 1
Router(config-keychain-key)# key-string CISCO
Router(config-keychain-key)# exit
Router(config)# oer master
Router(config-oer-mc)# border 10.100.1.1 key-chain OER
Router(config-oer-mc-br)# interface Ethernet 0/0 external
Router(config-oer-mc-br-if)# exit
Router(config-oer-mc-br)# interface Serial 1/1 internal
Router(config-oer-mc-br-if)# end
Router(config-oer-mc)# border 10.200.2.2 key-chain OER
Router(config-oer-mc-br)# interface Ethernet 2/2 external
Router(config-oer-mc-br-if)# exit
Router(config-oer-mc-br)# interface Serial 3/3 internal
Router(config-oer-mc-br-if)# end
```

BR1 Configuration

The following example, starting in global configuration mode, shows the configuration for BR1. eBGP peering is established with ISP1 (192.168.1.1 AS2). Standard community exchange and iBGP peering are established with BR2 (10.200.2.2) and internal peers (in the 10.150.1.0/24 network).

```
Router(config)# key chain OER
Router(config-keychain)# key 1
Router(config-keychain-key)# key-string CISCO
Router(config-keychain-key)# exit
Router(config-keychain)# exit
Router(config)# oer border
Router(config-oer-br)# master 172.16.1.1 key-chain OER
Router(config-oer-br)# local Serial 1/1
Router(config-oer-br)# exit
Router(config)# router bgp 1
Router(config-router)# neighbor 192.168.1.1 remote-as 2
Router(config-router)# neighbor 10.200.2.2 remote-as 1
Router(config-router)# neighbor 10.150.1.1 remote-as 1
Router(config-router)# neighbor 10.150.1.2 remote-as 1
Router(config-router)# neighbor 10.150.1.3 remote-as 1
Router(config-router)# address-family ipv4 unicast
Router(config-router-af)# neighbor 192.168.1.1 activate
Router(config-router-af)# neighbor 10.200.2.2 activate
Router(config-router-af)# neighbor 10.200.2.2 send-community standard
Router(config-router-af)# neighbor 10.150.1.1 activate
Router(config-router-af)# neighbor 10.150.1.1 send-community standard
Router(config-router-af)# neighbor 10.150.1.2 activate
Router(config-router-af)# neighbor 10.150.1.2 send-community standard
Router(config-router-af)# neighbor 10.150.1.3 activate
Router(config-router-af)# neighbor 10.150.1.3 send-community standard
Router(config-router-af)# end
```

BR2 Configuration

The following example, starting in global configuration mode, shows the configuration for BR2. eBGP peering is established with ISP2 (192.168.2.2 AS1). Standard community exchange and iBGP peering is established with BR2 (10.100.1.1) and internal peers (in the 10.150.1.0/24 network).

```
Router(config)# key chain OER
Router(config-keychain)# key 1
Router(config-keychain-key)# key-string CISCO
Router(config-keychain-key)# end
Router(config)# oer border
Router(config-oer-br)# master 172.16.1.1 key-chain OER
Router(config-oer-br)# local Serial 1/1
Router(config-oer-br)# exit
Router(config)# router bgp 1
Router(config-router)# neighbor 192.168.2.2 remote-as 3
Router(config-router)# neighbor 10.100.1.1 remote-as 1
Router(config-router)# neighbor 10.150.1.1 remote-as 1
Router(config-router)# neighbor 10.150.1.2 remote-as 1
Router(config-router)# neighbor 10.150.1.3 remote-as 1
Router(config-router)# address-family ipv4 unicast
Router(config-router-af)# neighbor 192.168.2.2 activate
Router(config-router-af)# neighbor 10.200.2.2 activate
Router(config-router-af)# neighbor 10.200.2.2 send-community standard
Router(config-router-af)# neighbor 10.150.1.1 activate
Router(config-router-af)# neighbor 10.150.1.1 send-community standard
Router(config-router-af)# neighbor 10.150.1.2 activate
Router(config-router-af)# neighbor 10.150.1.2 send-community standard
Router(config-router-af)# neighbor 10.150.1.3 activate
Router(config-router-af)# neighbor 10.150.1.3 send-community standard
Router(config-router-af)# end
```

Internal Peer Configuration

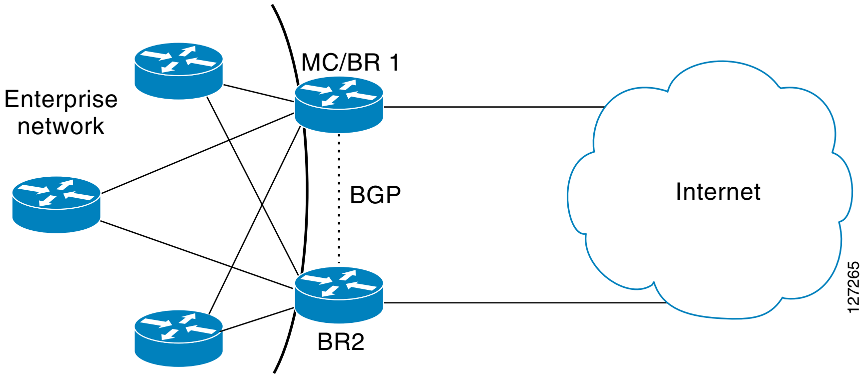
The following example, starting in global configuration mode, shows the internal peer configuration. Standard full-mesh iBGP peering is established with BR1 and BR2 and the internal peers in autonomous system 1.

```
Router(config)# router bgp 1
Router(config-router)# neighbor 10.100.1.1 remote-as 1
Router(config-router)# neighbor 10.200.2.2 remote-as 1
Router(config-router)# neighbor 10.150.1.1 remote-as 1
Router(config-router)# neighbor 10.150.1.2 remote-as 1
Router(config-router)# neighbor 10.150.1.3 remote-as 1
Router(config-router)# address-family ipv4 unicast
Router(config-router-af)# neighbor 10.100.1.1 activate
Router(config-router-af)# neighbor 10.100.1.1 send-community standard
Router(config-router-af)# neighbor 10.200.2.2 activate
Router(config-router-af)# neighbor 10.200.2.2 send-community standard
Router(config-router-af)# neighbor 10.150.1.1 activate
Router(config-router-af)# neighbor 10.150.1.1 send-community standard
Router(config-router-af)# neighbor 10.150.1.2 activate
Router(config-router-af)# neighbor 10.150.1.2 send-community standard
Router(config-router-af)# neighbor 10.150.1.3 activate
Router(config-router-af)# neighbor 10.150.1.3 send-community standard
Router(config-router-af)# end
```

OER Master Controller and Border Router Process Deployed on a Single Router with a Second Border Router: Example

Figure 10 shows an OER-managed network with two border routers. BR1 is configured to run a master controller and border router process.

Figure 10 OER Master Controller and Border Router Process Deployed on a Single Router with a Second Border Router



BR2 is configured as a border router. The internal network is running OSPF. Each border router peers with a different ISP. A static route to the egress interface is configured on each border router. The static routes are then redistributed into OSPF. Injected prefixes are advertised through static route redistribution.

BR1 Configuration: Master Controller and Border Router on a Single Router with Load Distribution Policy

The following example, starting in global configuration mode, shows the configuration of BR1. This router is configured to run both a master controller and a border router process. BR1 peers with ISP1. A traffic load distribution policy is configured under the master controller process that is applied to all exit links in the OER-managed network.

```
Router(config)# key chain OER
Router(config-keychain)# key 1
Router(config-keychain-key)# key-string CISCO
Router(config-keychain-key)# exit
Router(config-keychain)# exit
Router(config)# oer border
Router(config-oer-br)# master 10.100.1.1 key-chain OER
Router(config-oer-br)# local Loopback 0
Router(config-oer-br)# exit
Router(config)# oer master
Router(config-oer-mc)# logging
Router(config-oer-mc)# border 10.100.1.1 key-chain OER
Router(config-oer-mc-br)# interface Serial 0/0 external
Router(config-oer-mc-br-if)# exit
Router(config-oer-mc-br)# interface Ethernet 1/1 internal
Router(config-oer-mc-br-if)# exit
Router(config-oer-mc-br)# exit
Router(config-oer-mc)# border 10.200.2.2 key-chain OER
Router(config-oer-mc-br)# interface Serial 2/2 external
Router(config-oer-mc-br-if)# exit
Router(config-oer-mc-br)# interface Ethernet 3/3 internal
Router(config-oer-mc-br-if)# exit
Router(config-oer-mc-br)# exit
Router(config-oer-mc)# exit
Router(config)# ip route 0.0.0.0 0.0.0.0 Serial 0/0
Router(config)# !
Router(config)# route-map STATIC
Router(config-route-map)# match tag 5000
Router(config-route-map)# set metric -10
Router(config-route-map)# exit
Router(config)# router ospf 1
Router(config-router)# network 10.0.0.0 0.0.0.255 area 0
Router(config-router)# redistribute static route-map STATIC subnets
Router(config-router)# end
```

BR2 Configuration

The following example, starting in global configuration mode, shows the configuration of BR2. This router is configured to run only a border router process.

```
Router(config)# key chain OER
Router(config-keychain)# key 1
Router(config-keychain-key)# key-string CISCO
Router(config-keychain-key)# exit
Router(config-keychain)# exit
Router(config)# oer border
Router(config-oer-border)# master 10.100.1.1 key-chain OER
Router(config-oer-border)# local Ethernet3/3
Router(config-oer-border)# exit
Router(config)# ip route 0.0.0.0 0.0.0.0 Serial 2/2
Router(config)# !
Router(config)# route-map STATIC permit 10
Router(config-route-map)# match tag 5000
Router(config-route-map)# set metric -10
Router(config-route-map)# exit
Router(config)# router ospf 1
Router(config-router)# network 10.0.0.0 0.255.255.255 area 0
Router(config-router)# redistribute static route-map STATIC
Router(config-router)# end
```

Internal Peer Configuration

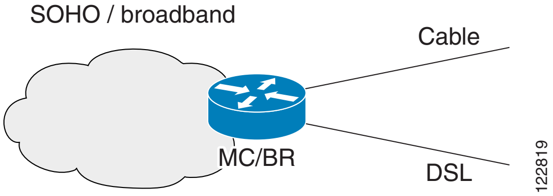
The following example, starting in global configuration mode, configures an OSPF routing process to establish peering with the border routers and internal peers. No redistribution is configured on the internal peers.

```
Router(config)# router ospf 1
Router(config-router)# network 10.0.0.0 0.255.255.255 area 0
Router(config-router)# redistribute static route-map STATIC subnets
Router(config-router)# end
```

OER Master Controller and Border Router Deployed on a Single Router: Example

Figure 11 shows a SOHO network in which the master controller and border router process are set up on a single router.

Figure 11 OER Deployed on a Single Router in a SOHO Configuration



The router connects the SOHO network with two ISPs. OER is configured to learn prefixes based on highest outbound throughput and lowest delay. Prefixes with a response time longer than 80 milliseconds are out-of-policy and moved if the performance on the other link conforms to the policy.

Master Controller and Border Router Configuration on a Single Router

The following example, starting in global configuration mode, shows an OER master controller and border router process deployed on a single router:

```
Router(config)# key chain OER
Router(config-keychain)# key 1
Router(config-keychain-key)# key-string KEYSTRING2
Router(config-keychain-key)# exit
Router(config-keychain)# exit
Router(config)# oer border
Router(config)# logging
Router(config-oer-br)# master 10.100.1.1 key-chain OER
Router(config-oer-br)# local Loopback 0
Router(config-oer-br)# exit
Router(config)# oer master
Router(config-oer-mc)# logging
Router(config-oer-mc)# border 10.100.1.1 key-chain OER
Router(config-oer-mc-br)# interface Ethernet 0/0 external
Router(config-oer-mc-br-if)# exit
Router(config-oer-mc-br)# interface Ethernet 1/1 external
Router(config-oer-mc-br-if)# exit
Router(config-oer-mc-br)# interface Ethernet 2/2 internal
Router(config-oer-mc-br-if)# exit
Router(config-oer-mc-br)# exit
Router(config-oer-mc)# exit
Router(config)# ip route 0.0.0.0 0.0.0.0 Ethernet 0/0
Router(config)# ip route 0.0.0.0 0.0.0.0 Ethernet 1/1
Router(config)# end
```

Registering an Application Interface Provider and Configuring Host Devices: Example

The following configuration example shows how to register a provider on a master controller. In this example, more than one provider is configured, so the priority is set for each provider. For the single host device configured for provider 1, no priority is set and the default priority value of 65535 is assigned giving this host device a lower priority than both the host devices configured for provider 2. After the provider is registered and an application on a host device initiates a session, some **show** commands can be entered on the master controller to help you track provider activity.

```
Router(config)# oer master
Router(config-oer-mc)# api provider 1 priority 3000
Router(config-oer-mc-api-provider)# host-address 10.1.2.2 key-chain OER_HOST
Router(config-oer-mc-api-provider)# exit
Router(config-oer-mc)# api provider 2 priority 4000
Router(config-oer-mc-api-provider)# host-address 10.2.2.2 key-chain OER_HOST
priority 3000
Router(config-oer-mc-api-provider)# host-address 10.2.2.3 key-chain OER_HOST
priority 4000
Router(config-oer-mc-api-provider)# end
!
Router# show oer api provider detail
Router# show oer master policy dynamic
Router# show oer master prefix 10.1.1.0/24 report
```

Where to Go Next

Now that your OER network components are set up, you should read through the other modules in the following order:

- Using OER to Profile the Traffic Classes
- Measuring the Traffic Class Performance and Link Utilization Using OER
- Configuring and Applying OER Policies
- Using OER to Control Traffic Classes and Verify the Route Control Changes

After you understand the various OER phases, review the OER solutions modules that are listed under "Related Documents" section on page 58.

Additional References

The following sections provide references related to setting up OER network components.

Related Documents

Related Topic	Document Title
Cisco OER technology overview	"Cisco IOS Optimized Edge Routing Overview" module
OER solution module: voice traffic optimization using OER active probes.	"OER Voice Traffic Optimization Using Active Probes" module
OER solution module: configuring VPN IPsec/GRE tunnel interfaces as OER-managed exit links.	"Configuring VPN IPsec/GRE Tunnel Interfaces As OER-Managed Exit Links" module
Cisco OER commands: complete command syntax, command mode, command history, defaults, usage guidelines and examples	*Cisco IOS Optimized Edge Routing Command Reference*
IP Routing Protocol commands	*Cisco IOS IP Routing Protocols Command Reference*
Key Chain Authentication: information about authentication key configuration and management in Cisco IOS software	"Managing Authentication Keys" section of the "Configuring IP Routing Protocol-Independent Features" chapter in the *Cisco IOS IP Routing Protocols Configuration Guide*

Standards

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified.	—

MIBs

MIB	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported, and support for existing RFCs has not been modified.	—

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Setting Up OER Network Components

Table 3 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.3(8)T, 12.2(33)SRB, or a later release appear in the table.

For information on a feature in this technology that is not documented here, see the "Cisco IOS Optimized Edge Routing Features Roadmap."

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 3 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 3 Feature Information for Setting Up OER Network Components

Feature Name	Releases	Feature Configuration Information
Optimized Edge Routing	12.3(8)T 12.2(33)SRB	OER was introduced.
Automatic Port Configuration [1]	12.3(11)T 12.2(33)SRB	Support for automatic port configuration was introduced. Communication between the master controller and border router is automatically carried over port 3949 when connectivity is established. Port 3949 is registered with IANA for OER communication. Manual port number configuration is required only if you are running Cisco IOS Release 12.3(8)T or if you need to configure OER communication to use a dynamic port number. The following section provides information about this feature: • Setting Up the OER Master Controller, page 15 No commands were introduced by this feature.
Support for NAT and Static Routing[2]	12.3(14)T 12.2(33)SRB	Support to allow OER to control traffic class routing using static routing in networks using NAT. The following sections provide information about this feature: • OER and NAT, page 11 • Configuring OER to Control Traffic with Static Routing in Networks Using NAT, page 28 • Configuring OER to Control Traffic with Static Routing in Networks Using NAT: Example, page 48 The following command was modified by this feature: **ip nat inside source**.
Support for VLAN Interfaces[3]	12.3(14)T 12.2(33)SRB	Support to configure a VLAN interface as an internal interface was introduced. The following section provides information about this feature: • Setting Up the OER Master Controller, page 15 No commands were introduced by this feature.

Table 3 ***Feature Information for Setting Up OER Network Components (continued)***

Feature Name	Releases	Feature Configuration Information
Application Performance Routing: PBR	12.4(2)T 12.2(33)SRB	The Application Performance Routing: PBR feature introduces the capability to optimize IP traffic based on the type of application that is carried by the monitored prefix. Independent policy configuration is applied to the subset (application) of traffic. The ability to configure a Cisco router as an interim border router to allow a border router to be more than one hop away from the master controller, was also introduced. The following sections provide information about this feature: • Single Hop Peer Restriction Avoidance using OER Interim Border Routers, page 5 • Configuring an Interim Border Router, page 25 • Configuring an Interim Border Router: Example, page 48 The following commands were introduced or modified by this feature: **debug oer border pbr**, **debug oer master prefix**, **match ip address (OER)**, **show oer master active-probes**, and **show oer master appl**.

Table 3 Feature Information for Setting Up OER Network Components (continued)

Feature Name	Releases	Feature Configuration Information
Performance Routing - Application Interface	12.4(15)T	The Performance Routing - Application Interface feature introduces support for an OER application interface. The application interface defines the mode of communication and messaging between applications and the network for the purpose of optimizing the traffic associated with the applications. A provider must be registered with an OER master controller before the application can interface with OER. Host devices in the provider network running an application that communicates with OER using the application interface must also be configured at an OER master controller with an IP address and key chain password. The following sections provide information about this feature: • OER Application Interface, page 12 • Registering an Application Interface Provider and Configuring Host Devices, page 42 • Displaying Information about Application Interface Provider Activity, page 44 • Registering an Application Interface Provider and Configuring Host Devices: Example, page 56 The following commands were introduced or modified by this feature: **api provider**, **debug oer api**, **host-address**, **show oer api provider**, **show oer master policy**, and **show oer master prefix**.

Table 3 Feature Information for Setting Up OER Network Components (continued)

Feature Name	Releases	Feature Configuration Information
OER Border Router Only Functionality	12.2(33)SXH	In Cisco IOS Release 12.2(33)SXH support for using a Cisco Catalyst 6500 series switch as an OER border router was introduced. Only border router functionality is included in the Cisco IOS Release 12.2(33)SXH images; no master controller configuration is available. The master controller that communicates with the Cisco Catalyst 6500 series switch being used as a border router must be a router running Cisco IOS Release 12.4(6)T or a later release. The OER master controller software has been modified to handle the limited functionality supported by the Cisco Catalyst 6500 border routers. Using the Route Processor (RP), the Catalyst 6500 border routers can capture throughput statistics only for a traffic class compared to the delay, loss, unreachability, and throughput statistics collected by non-Catalyst 6500 border routers. A master controller automatically detects the limited capabilities of the Catalyst 6500 border routers and downgrades other border routers to capture only the throughput statistics for traffic classes. By ignoring other types of statistics, the master controller is presented with a uniform view of the border router functionality. The following sections provide information about this feature: • OER Border Router Support for Cisco Catalyst 6500 Series Switches, page 5 • Setting Up an OER Border Router, page 21 The following command was introduced or modified by this feature: **show oer border passive cache**.

1. This is a minor enhancement. Minor enhancements are not typically listed in Feature Navigator.
2. This is a minor enhancement. Minor enhancements are not typically listed in Feature Navigator.
3. This is a minor enhancement. Minor enhancements are not typically listed in Feature Navigator.

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

Using OER to Profile the Traffic Classes

First Published: January 29, 2007
Last Updated: July 11, 2008

This module describes how Optimized Edge Routing (OER) profiles the traffic classes. To optimize traffic routing, subsets of the total traffic must be identified, and these traffic subsets are named traffic classes. The OER master controller can profile traffic classes either by manual configuration on the master controller, or by automatic learning on the basis of parameters such as throughput or delay characteristics of traffic on the border routers. Automatic learning requires traffic class parameters to be configured on the master controller.

Note If you are running Cisco IOS Release 12.4(15)T or a later release, please refer to the "Using Performance Routing to Profile the Traffic Classes" module for the latest configuration information and tasks introduced for Performance Routing. Performance Routing (PfR) is an extension of the Optimized Edge Routing (OER) technology and the commands and command modes for PfR use the oer naming convention.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Using OER to Profile the Traffic Classes" section on page 40.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Contents

Prerequisites for Using OER to Profile the Traffic Classes

- Before implementing the OER profile phase, you need to understand an overview of how OER works and how to set up OER network components. See the "Cisco IOS Optimized Edge Routing Overview" and "Setting Up OER Network Components" modules for more details.
- Cisco Express Forwarding (CEF) must be enabled on all participating devices. No other switching path is supported, even if otherwise supported by PBR.

Restrictions for Using OER to Profile the Traffic Classes

If any of the border routers is a Cisco Catalyst 6500 switch and the master controller has set the monitoring mode to special, only the throughput method of learning is used to profile the traffic classes. If both delay and throughput are configured, the master controller will ignore the delay configuration. For more details about the special monitoring mode, see the "Measuring the Traffic Class Performance and Link Utilization Using OER" module for more details.

Information About Using OER to Profile the Traffic Classes

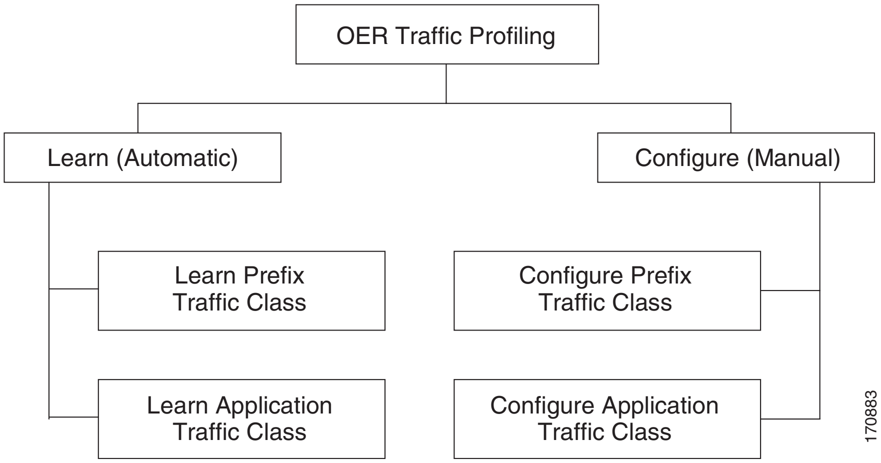
To configure the master controller to profile traffic classes, you should understand the following concepts:

- OER Traffic Class Profiling, page 2
- OER Automatic Traffic Class Learning, page 3
- OER Manual Traffic Class Configuration, page 6

OER Traffic Class Profiling

Before optimizing traffic, OER has to determine the traffic classes from the traffic flowing through the border routers. To optimize traffic routing, subsets of the total traffic must be identified, and these traffic subsets are named traffic classes. The list of traffic classes entries is named a Monitored Traffic Class (MTC) list. The entries in the MTC list can be profiled either by automatically learning the traffic flowing through the device or by manually configuring the traffic classes. Learned and configured traffic classes can both exist in the MTC list at the same time. The OER profile phase includes both the learn mechanism and the configure mechanism. The overall structure of the OER traffic class profile process and its component parts can be seen in Figure 1.

Figure 1 OER Traffic Class Profiling Process


OER Traffic Profiling
Learn (Automatic)
Configure (Manual)
Learn Prefix Traffic Class
Configure Prefix Traffic Class
Learn Application Traffic Class
Configure Application Traffic Class
170883

The ultimate objective of this phase is to select a subset of traffic flowing through the network. This subset of traffic—the traffic classes in the MTC list—represents the classes of traffic that need to be routed based on the best performance path available.

More details about each of the components in Figure 1 are contained in the following concepts:

- OER Automatic Traffic Class Learning, page 3
- OER Manual Traffic Class Configuration, page 6

OER Automatic Traffic Class Learning

OER can automatically learn the traffic classes while monitoring the traffic flow through border routers. Although the goal is to optimize a subset of the traffic, you may not know all the exact parameters of this traffic and OER provides a method to automatically learn the traffic and create traffic classes by populating the MTC list. Several features have been added to OER since the original release to add functionality to the automatic traffic class learning process.

Within the automatic traffic class learning process there are now three components. One component describes the automatic learning of prefix-based traffic classes, the second component describes automatic learning of application-based traffic classes, and the third component describes the use of learn lists to categorize both prefix-based and application-based traffic classes. These three components are described in the following sections:

- Prefix Traffic Class Learning Using OER, page 4
- Application Traffic Class Learning Using OER, page 4
- Learn List Configuration Mode, page 5

Prefix Traffic Class Learning Using OER

The OER master controller can be configured, using NetFlow Top Talker functionality, to automatically learn prefixes based on the highest outbound throughput or the highest delay time. Throughput learning measures prefixes that generate the highest outbound traffic volume. Throughput prefixes are sorted from highest to lowest. Delay learning measures prefixes with the highest round-trip response time (RTT) to optimize these highest delay prefixes to try to reduce the RTT for these prefixes. Delay prefixes are sorted from the highest to the lowest delay time.

OER can automatically learn two types of prefixes:

- outside prefix—An outside prefix is defined as a public IP prefix assigned outside the company. Outside prefixes are received from other networks.
- inside prefix—An inside prefix is defined as a public IP prefix assigned to a company. An inside prefix is a prefix configured within the company network.

In Cisco IOS Release 12.4(9)T, 12.2(33)SRB, and later releases, the ability to learn inside prefixes was introduced. Using BGP, OER can select inside prefixes to support best entrance selection for traffic that originates from prefixes outside an autonomous system destined for prefixes inside the autonomous system. In prior releases, only outside prefixes were supported. Company networks advertise the inside prefixes over the Internet using an Internet service provider (ISP) and receive advertisements for outside prefixes from an ISP.

Note Although OER can learn an inside prefix, OER will not try to control an inside prefix unless there is an exact match in the BGP routing information base (RIB) because OER does not advertise a new prefix to the Internet.

Automatic prefix learning is configured in OER Top Talker and Top Delay learning configuration mode. The **learn** command is used to enter this mode from OER master controller configuration mode. When automatic prefix learning is enabled, prefixes and their delay or throughput characteristics are measured on the border routers. Performance measurements for the prefix-based traffic classes are reported to the master controller where the learned prefixes are stored in the MTC list.

Prefixes are learned on the border routers through monitoring the traffic flow using the embedded NetFlow capability. All incoming and outgoing traffic flows are monitored. The top 100 flows are learned by default, but the master controller can be configured to learn up to 2500 flows for each learn cycle. The master controller can control a maximum of 5000 prefixes.

The master controller can be configured to aggregate learned prefixes based on type, BGP or non-BGP (static). Prefixes can be aggregated based on the prefix length. Traffic flows are aggregated using a /24 prefix length by default. Prefix aggregation can be configured to include any subset or superset of the network, from single host route (/32) to a major network address range. For each aggregated prefix, up to five host addresses are selected to use as active probe targets. Prefix aggregation is configured with the **aggregation-type** command in OER Top Talker and Delay learning configuration mode.

Application Traffic Class Learning Using OER

In the first release of OER, Cisco IOS Release 12.3(8)T, only Layer 3 prefixes could be learned. In subsequent releases, Layer 4 options such as protocol or port numbers were added as filters to the prefix-based traffic class. The protocol and port numbers can be used to identify specific application traffic classes; protocol and port number parameters are monitored only within the context of a prefix and are not sent to the master controller database (MTC list). The prefix that carries the specific traffic is then monitored by the master controller. In Cisco IOS Release 12.4(9)T, Release 12.2(33)SRB, and later releases, application traffic class learning supports Differentiated Services Code Point (DSCP) values in addition to protocol and port numbers, and these Layer 4 options are entered in the MTC list.

Port and Protocol Based Prefix Learning by OER

In Cisco IOS Release 12.3(11)T, Release 12.2(33)SRB, and later releases, prefix learning on the basis of port numbers or protocols was introduced. This feature allows you to configure the master controller to filter the prefix-based traffic class based on the protocol number or the source or destination port number, carried by TCP or UDP traffic. This feature provides a very granular filter that can be used to further optimize prefixes learned based on throughput and delay. The traffic classes sent to the MTC list on the master controller, however, only contain the prefix information, not the protocol and port numbers.

Port and protocol based prefix learning allows you to optimize or exclude traffic streams for a specific protocol or the TCP port, UDP port, or range of port numbers. Traffic can be optimized for a specific application or protocol. Uninteresting traffic can be excluded, allowing you to focus router system resources, and reduce unnecessary CPU and memory utilization. In cases where traffic streams need to be excluded or included over ports that fall above or below a certain port number, the range of port numbers can be specified. Port and protocol prefix based learning is configured with the **protocol** command in OER Top Talker and Top Delay learning configuration mode.

For a list of IANA assigned port numbers, see the following document:

- http://www.iana.org/assignments/port-numbers

For a list of IANA assigned protocol numbers, see the following document:

- http://www.iana.org/assignments/protocol-numbers

DSCP Value, Port, and Protocol Learning by OER

In Cisco IOS Release 12.4(9)T, 12.2(33)SRB, and later releases, the ability to filter and aggregate application traffic by DSCP value, port number or protocol was introduced. Traffic classes can be defined by a combination of keys comprising of protocol, port numbers, and DSCP values. The ability to filter out traffic that is not required, and the ability to aggregate the traffic in which you are interested, was introduced. Information such as protocol, port number, and DSCP value is now sent to the master controller database in addition to the prefix information. The new functionality allows OER to both actively and passively monitor application traffic. Using new CLI and access lists, OER can be configured to automatically learn application traffic classes.

Learn List Configuration Mode

In Cisco IOS Release 12.4(15)T, a new configuration mode named learn list was introduced. Learn lists are a way to categorize learned traffic classes. In each learn list, different criteria including prefixes, application definitions, filters, and aggregation parameters for learning traffic classes can be configured.

If you are running Cisco IOS Release 12.4(15)T or a later release, please refer to the "Using Performance Routing to Profile the Traffic Classes" module for learn list configuration information and tasks introduced for Performance Routing. Performance Routing (PfR) is an extension of the Optimized Edge Routing (OER) technology and the commands and command modes for PfR use the oer naming convention.

OER Manual Traffic Class Configuration

OER can be manually configured to create traffic classes for monitoring and subsequent optimizing. Automatic learning generally uses a default prefix length of /24 but manual configuration allows exact prefixes to be defined. Within the manual traffic class configuration process there are two components—manually configuring prefix-based traffic classes and manually configuring application-based traffic classes, both of which are described in the following sections:

- Prefix Traffic Class Configuration Using OER, page 6
- Application Traffic Class Configuration Using OER, page 7

Prefix Traffic Class Configuration Using OER

A prefix or range of prefixes can be selected for OER monitoring by configuring an IP prefix list. The IP prefix list is then imported into the MTC list by configuring a match clause in an OER map. An OER map is similar to an IP route map. IP prefix lists are configured with the **ip prefix-list** command and OER maps are configured with the **oer-map** command in global configuration mode.

The prefix list syntax operates in a slightly different way with OER than in regular routing. The **ge** keyword is not used and the **le** keyword is used by OER to specify only an inclusive prefix. A prefix list can also be used to specify an exact prefix.

A master controller can monitor and control an exact prefix of any length including the default route. If an exact prefix is specified, OER monitors only the exact prefix.

A master controller can monitor and control an inclusive prefix using the **le** keyword and the *le-value* argument set to 32. OER monitors the configured prefix and any more specific prefixes (for example, configuring the 10.0.0.0/8 le 32 prefix would include the 10.1.0.0/16 and the 10.1.1.0/24 prefixes) over the same exit and records the information in the routing information base (RIB).

Note Use the inclusive prefix option with caution in a typical OER deployment because of the potential increase in the amount of prefixes being monitored and recorded.

An IP prefix list with a deny statement can be used to configure the master controller to exclude a prefix or prefix length for learned traffic classes. Deny prefix list sequences should be applied in the lowest OER map sequences for best performance. In Cisco IOS Release 12.4(9)T, 12.2(33)SRB, and later releases, the master controller can be configured to tell border routers to filter out uninteresting traffic using an access list.

Note IP prefix lists with deny statements can be applied only to learned traffic classes.

Two types of prefix can be manually configured for OER monitoring using an IP prefix list:

- outside prefix—An outside prefix is defined as a public IP prefix assigned outside the company. Outside prefixes are received from other networks.
- inside prefix—An inside prefix is defined is defined as a public IP prefix assigned to a company. An inside prefix is a prefix configured within the company network.

In Cisco IOS Release 12.4(9)T, 12.2(33)SRB, and later releases, the ability to manually configure inside prefixes was introduced. Using BGP, OER can be configured to select inside prefixes to support best entrance selection for traffic that originates from prefixes outside an autonomous system destined for prefixes inside the autonomous system. In prior releases, only outside prefixes were supported. Company networks advertise the inside prefixes over the Internet using an Internet service provider (ISP) and receive advertisements for outside prefixes from an ISP.

Note Although an inside prefix can be manually configured for OER monitoring, OER will not try to control an inside prefix unless there is an exact match in the BGP routing information base (RIB) because OER does not advertise a new prefix to the Internet.

Application Traffic Class Configuration Using OER

In the first release of OER, Cisco IOS Release 12.3(8)T, only Layer 3 prefixes could be manually configured during the OER profile phase. In Cisco IOS Release 12.4(2)T, 12.2(33)SRB, and later releases, support for OER application-aware routing for policy-based routing (PBR) was introduced. Application-aware routing allows the selection of traffic for specific applications based on values in the IP packet header, other than the Layer 3 destination address through a named extended IP access control list (ACL). Only named extended ACLs are supported. The extended ACL is configured with a permit statement and then referenced in an OER map. The protocol and port numbers can be used to identify specific application traffic classes, but protocol and port number parameters are monitored only within the context of a prefix, and are not sent to the MTC list. Only the prefix that carries the specific application traffic is profiled by the master controller. With application-aware routing support, active monitoring of application traffic was supported. Passive monitoring of application traffic was introduced in Cisco IOS Release 12.4(9)T, 12.2(33)SRB, and later releases, with application traffic class configuration support of the profiling of DSCP values as well as protocol and port numbers. DSCP values, port numbers, and protocols in addition to prefixes, are all now stored in the MTC list.

In Cisco IOS Release 12.4(15)T, new static application mapping was introduced under OER map configuration mode to simplify the configuration of traffic classes. If you are running Cisco IOS Release 12.4(15)T or a later release, please refer to the "Using Performance Routing to Profile the Traffic Classes" module for static application mapping configuration information and tasks introduced for Performance Routing. Performance Routing (PfR) is an extension of the Optimized Edge Routing (OER) technology and the commands and command modes for PfR use the oer naming convention.

How to Configure OER to Profile the Traffic Classes

An OER master controller can be configured to automatically learn the traffic classes, or the traffic classes can be manually configured. Two types of traffic classes—to be automatically learned or manually configured—can be profiled:

- Traffic classes based on destination prefixes
- Traffic classes representing custom application definitions using access lists

Note In Cisco IOS Release 12.4(15)T, the introduction of learn lists allows traffic classes that are automatically learned by OER to be categorized into separate learn lists to which different OER policies can be applied. If you are running Cisco IOS Release 12.4(15)T or a later release, please refer to the "Using Performance Routing to Profile the Traffic Classes" module for learn list configuration information and tasks introduced for Performance Routing. Performance Routing (PfR) is an extension of the Optimized Edge Routing (OER) technology and the commands and command modes for PfR use the oer naming convention.

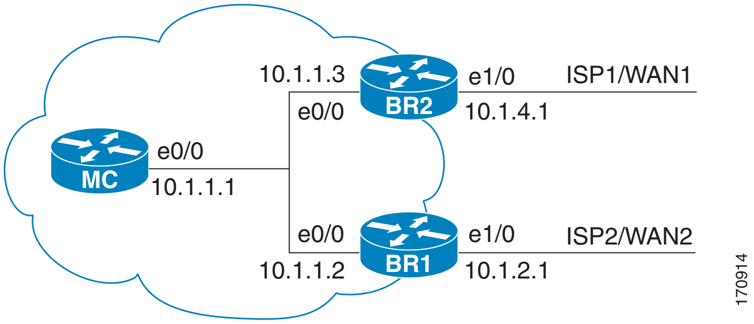
One or more of the following tasks may be performed:

- Configuring OER to Automatically Learn Prefix-Based Traffic Classes, page 8
- Configuring OER to Automatically Learn Traffic Classes Using Inside Prefixes, page 11
- Configuring OER to Automatically Learn Prefix-Based Traffic Classes Using Protocol or Port Number, page 13
- Specifying the Flow Keys for Automatic Learning of Application Traffic Classes, page 17
- Creating an Access List to Specify a Filter for Automatically Learned Application Traffic, page 19
- Creating an Access List to Specify Aggregation Criteria for Automatically Learned Application Traffic, page 23
- Displaying Application Traffic Flow Information on a Border Router, page 27
- Manually Selecting Prefixes for OER Monitoring, page 29
- Manually Selecting Inside Prefixes for OER Monitoring, page 31
- Manually Selecting Traffic Classes Using Prefix, Protocol, Port, and DSCP Value, page 33

Configuring OER to Automatically Learn Prefix-Based Traffic Classes

Perform this task to configure an OER master controller to automatically learn prefixes to be used as traffic classes to be entered in the MTC list. This task is performed on the master controller shown in Figure 2.

Figure 2 Network Diagram of OER Master Controller and Border Routers



The **learn** command is entered in OER master controller configuration mode and is required to enter OER Top Talker and Top Delay configuration mode. This task configures prefix learning based on the highest outbound throughput or the highest delay time, and one or both of these parameters must be specified. Optional configuration parameters such as learning period timers, maximum number of prefixes, and an expiration time for MTC list entries are also shown.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **oer master**
4. **learn**
5. **delay**

6. **throughput**
7. **aggregation-type** {**bgp** | **non-bgp** | **prefix-length** *prefix-mask*}
8. **monitor-period** *minutes*
9. **periodic-interval** *minutes*
10. **prefixes** *number*
11. **expire after** {**session** *number* | **time** *minutes*}
12. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	`enable`
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	`configure terminal`
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	`oer master`
	Example:
	`Router(config)# oer master`
		Enters OER master controller configuration mode to configure a Cisco router as a master controller and to configure master controller policy and timer settings.

Step 4	`learn`
	Example:
	`Router(config-oer-mc)# learn`
		Enters OER Top Talker and Top Delay learning configuration mode to configure prefix learning and timers.

Step 5	`delay`
	Example:
	`Router(config-oer-mc-learn)# delay`
		Enables prefix learning based on the highest delay time.
		• *Top Delay* prefixes are sorted from the highest to lowest delay time.
		• The example configures prefix learning based on the highest delay.
		Note To configure OER learning you must specify either the **delay** command, the **throughput** command, or both commands.

Step 6	`throughput`
	Example:
	`Router(config-oer-mc-learn)# throughput`
		Configures the master controller to learn the top prefixes based on the highest outbound throughput.
		• When this command is enabled, the master controller will learn the top prefixes across all border routers according to the highest outbound throughput.
		• The example configures a master controller to learn the top prefixes based on highest outbound throughput.

	Command or Action	Purpose
Step 7	**aggregation-type** {**bgp** | **non-bgp** | **prefix-length**} *prefix-mask* **Example:** `Router(config-oer-mc-learn)# aggregation-type bgp`	(Optional) Configures a master controller to aggregate learned prefixes based on traffic flow type. • The **bgp** keyword configures prefix aggregation based on entries in the BGP routing table. This keyword is used if BGP peering is enabled in the network. • The **non-bgp** keyword configures learned prefix aggregation based on static routes. Entries in the BGP routing table are ignored when this keyword is entered. • The **prefix-length** keyword configures aggregation based on the specified prefix length. The range of values that can be configured for this argument is a prefix mask from 1 to 32. • If this command is not specified, the default aggregation is performed based on a /24 prefix length. • The example configures BGP prefix aggregation.
Step 8	**monitor-period** *minutes* **Example:** `Router(config-oer-mc-learn)# monitor-period 10`	(Optional) Sets the time period that an OER master controller learns traffic flows. • The default learning period is 5 minutes. • The length of time between monitoring periods is configured with the **periodic-interval** command. • The number of prefixes that are learned is configured with the **prefixes** command. • The example sets the length of each monitoring period to 10 minutes.
Step 9	**periodic-interval** *minutes* **Example:** `Router(config-oer-mc-learn)# periodic-interval 20`	(Optional) Sets the time interval between prefix learning periods. • By default, the interval between prefix learning periods is 120 minutes. • The example sets the time interval between monitoring periods to 20 minutes.
Step 10	**prefixes** *number* **Example:** `Router(config-oer-mc-learn)# prefixes 200`	(Optional) Sets the number of prefixes that the master controller will learn during the monitoring period. • By default, the top 100 traffic flows are learned. • The example configures a master controller to learn 200 prefixes during each monitoring period.

	Command or Action	Purpose

Step 11	**expire after** {**session** *number* | **time** *minutes*}
	Example:
	`Router(config-oer-mc-learn)# expire after session 100`
		(Optional) Sets the length of time that learned prefixes are kept in the central policy database.
		• The **session** keyword configures learned prefixes to be removed after the specified number of monitoring periods have occurred.
		• The **time** keyword configures learned prefixes to be removed after the specified time period. The time value is entered in minutes.
		• The example configures learned prefixes to be removed after 100 monitoring periods.

Step 12	**end**
	Example:
	`Router(config-oer-mc)# end`
		Exits OER Top Talker and Top Delay learning configuration mode, and returns to privileged EXEC mode.

What to Do Next

This section shows how to configure automatic prefix learning. To configure specific prefixes for OER monitoring and optimization, see the "Manually Selecting Prefixes for OER Monitoring" section on page 29.

Configuring OER to Automatically Learn Traffic Classes Using Inside Prefixes

In Cisco IOS Release 12.4(9)T, 12.2(33)SRB, and later releases, the OER BGP inbound optimization feature introduced the ability to automatically learn inside prefixes to support best entrance selection for traffic that originates from prefixes outside an autonomous system destined for prefixes inside the autonomous system.

Perform this task to configure an OER master controller to automatically learn inside prefixes to be used as traffic classes to be entered in the MTC list. This task is configured at the master controller and introduces the **inside bgp** command used in OER Top Talker and Top Delay configuration mode. This task configures automatic prefix learning of the inside prefixes (prefixes within the network). Optional configuration parameters such as learning period timers, maximum number of prefixes, and an expiration time for MTC list entries are also shown.

Prerequisites

- Before configuring this task, BGP peering for internal and external BGP neighbors must be configured.
- This task requires Cisco IOS Release 12.4(9)T, 12.2(33)SRB, or later release to be running on the master controller and border routers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **oer master**
4. **learn**
5. **inside bgp**
6. **monitor-period** *minutes*
7. **periodic-interval** *minutes*
8. **prefixes** *number*
9. **expire after** {**session** *number* | **time** *minutes*}
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**oer master** **Example:** `Router(config)# oer master`	Enters OER master controller configuration mode to configure a router as a master controller and to configure global operations and policies.
Step 4	**learn** **Example:** `Router(config-oer-mc)# learn`	Enters OER Top Talker and Top Delay learning configuration mode to configure prefix learning policies and timers.
Step 5	**inside bgp** **Example:** `Router(config-oer-mc-learn)# inside bgp`	Learns prefixes inside the network.
Step 6	**monitor-period** *minutes* **Example:** `Router(config-oer-mc-learn)# monitor-period 10`	(Optional) Sets the time period that an OER master controller learns traffic flows. • The default learning period is 5 minutes. • The length of time between monitoring periods is configured with the **periodic-interval** command. • The number of prefixes that are learned is configured with the **prefixes** command. • The example sets the length of each monitoring period to 10 minutes.

	Command or Action	Purpose
Step 7	**periodic-interval** *minutes* **Example:** `Router(config-oer-mc-learn)# periodic-interval 20`	(Optional) Sets the time interval between prefix learning periods. • By default, the interval between prefix learning periods is 120 minutes. • The example sets the time interval between monitoring periods to 20 minutes.
Step 8	**prefixes** *number* **Example:** `Router(config-oer-mc-learn)# prefixes 200`	(Optional) Sets the number of prefixes that the master controller will learn during the monitoring period. • By default, the top 100 traffic flows are learned. • The example configures a master controller to learn 200 prefixes during each monitoring period.

Step 9	**expire after** {**session** *number* | **time** *minutes*}
	Example:
	`Router(config-oer-mc-learn)# expire after session 100`
		(Optional) Sets the length of time that learned prefixes are kept in the central policy database.
		• The **session** keyword configures learned prefixes to be removed after the specified number of monitoring periods have occurred.
		• The **time** keyword configures learned prefixes to be removed after the specified time period. The time value is entered in minutes.
		• The example configures learned prefixes to be removed after 100 monitoring periods.

Step 10	**end**
	Example:
	`Router(config-oer-mc-learn)# end`
		Exits OER Top Talker and Top Delay learning configuration mode, and enters privileged EXEC mode.

What to Do Next

This section shows how to configure automatic prefix learning for inside prefixes. To configure specific inside prefixes for OER monitoring and optimization, see the "Manually Selecting Inside Prefixes for OER Monitoring" section on page 31.

Configuring OER to Automatically Learn Prefix-Based Traffic Classes Using Protocol or Port Number

Perform this task to configure an OER master controller to learn traffic classes to be entered in the MTC list based on prefixes but filtered by the protocol or port number. This task is performed on a master controller. The **learn** command is entered in OER master controller configuration mode and is required to enter OER Top Talker and Top Delay configuration mode. This task configures prefix learning based on the highest outbound throughput or the highest delay time and one or both of these parameters must be specified. After the prefix has been learned, a protocol or port number can be specified to create a subset of traffic classes. Optional configuration parameters such as learning period timers, the maximum number of prefixes, and an expiration time for MTC list entries are also shown.

Prerequisites

This task requires Cisco IOS Release 12.3(11)T, 12.2(33)SRB, or later release, to be running on the master controller and border routers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **oer master**
4. **learn**
5. **delay**
6. **throughput**
7. **aggregation-type** {**bgp** | **non-bgp** | **prefix-length** *prefix-mask*}
8. **monitor-period** *minutes*
9. **periodic-interval** *minutes*
10. **prefixes** *number*
11. **expire after** {**session** *number* | **time** *minutes*}
12. **protocol** {*number* | **tcp** | **udp**} [**port** *port-number* | **gt** *port-number* | **lt** *port-number* | **range** *lower-number upper-number*] [**dst** | **src**]
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**oer master** **Example:** `Router(config)# oer master`	Enters OER master controller configuration mode to configure a Cisco router as a master controller and to configure master controller policy and timer settings.
Step 4	**learn** **Example:** `Router(config-oer-mc)# learn`	Enters OER Top Talker and Top Delay learning configuration mode to configure prefix learning policies and timers.

	Command or Action	Purpose

Step 5	**`delay`**
	Example:
	`Router(config-oer-mc-learn)# delay`
		Enables prefix learning based on the highest delay time.
		• *Top Delay* prefixes are sorted from the highest to lowest delay time.
		• The example configures prefix learning based on the highest delay.
		Note To configure OER learning you must specify either the **delay** command, the **throughput** command, or both commands.

Step 6	**`throughput`**
	Example:
	`Router(config-oer-mc-learn)# throughput`
		Configures the master controller to learn the top prefixes based on the highest outbound throughput.
		• When this command is enabled, the master controller will learn the top prefixes across all border routers according to the highest outbound throughput.
		• The example configures a master controller to learn the top prefixes based on highest outbound throughput.

Step 7	**`aggregation-type`** {**`bgp`** | **`non-bgp`** | **`prefix-length`**} *`prefix-mask`*
	Example:
	`Router(config-oer-mc-learn)# aggregation-type bgp`
		(Optional) Configures a master controller to aggregate learned prefixes based on traffic flow type.
		• The **bgp** keyword configures prefix aggregation based on entries in the BGP routing table. This keyword is used if BGP peering is enabled in the network.
		• The **non-bgp** keyword configures learned prefix aggregation based on static routes. Entries in the BGP routing table are ignored when this keyword is entered.
		• The **prefix-length** keyword configures aggregation based on the specified prefix length. The range of values that can be configured for this argument is a prefix mask from 1 to 32.
		• If this command is not specified, the default aggregation is performed based on a /24 prefix length.
		• Up to five host addresses are learned for active monitoring when a prefix is aggregated.
		• The example configures BGP prefix aggregation.

Step 8	**`monitor-period`** *`minutes`*
	Example:
	`Router(config-oer-mc-learn)# monitor-period 10`
		(Optional) Sets the time period that an OER master controller learns traffic flows.
		• The default learning period is 5 minutes.
		• The length of time between monitoring periods is configured with the **periodic-interval** command.
		• The number of prefixes that are learned is configured with the **prefixes** command.
		• The example sets the length of each monitoring period to 10 minutes.

	Command or Action	Purpose
Step 9	**periodic-interval** *minutes* **Example:** `Router(config-oer-mc-learn)# periodic-interval 20`	(Optional) Sets the time interval between prefix learning periods. • By default, the interval between prefix learning periods is 120 minutes. • The example sets the time interval between monitoring periods to 20 minutes.
Step 10	**prefixes** *number* **Example:** `Router(config-oer-mc-learn)# prefixes 200`	(Optional) Sets the number of prefixes that the master controller will learn during the monitoring period. • By default, the top 100 traffic flows are learned. • The example configures a master controller to learn 200 prefixes during each monitoring period.

Step 11	**expire after** {**session** *number* | **time** *minutes*}
	Example:
	`Router(config-oer-mc-learn)# expire after session 100`
		(Optional) Sets the length of time that learned prefixes are kept in the central policy database.
		• The **session** keyword configures learned prefixes to be removed after the specified number of monitoring periods have occurred.
		• The **time** keyword configures learned prefixes to be removed after the specified time period. The time value is entered in minutes.
		• The example configures learned prefixes to be removed after 100 monitoring periods.

Step 12	**protocol** {*protocol-number* | **tcp** | **udp**} [**port** *port-number* | **gt** *port-number* | **lt** *port-number* | **range** *lower-number upper-number*] [**dst** | **src**] **Example:** `Router(config-oer-mc-learn)# protocol tcp port range 49542 49478`	Configures the master controller to learn prefixes based on a protocol number, TCP or UDP port number, or a range of port numbers. • Filtering based on a specific protocol is configured with the *protocol-number* argument. • TCP or UDP based filtering is enabled by configuring the **tcp** or **udp** keyword. • Port based filtering is enabled by configuring the **port** keyword. Port number ranges can be filtered based on greater-than or equal-to and less-than or equal-to filtering, or can be filtered by specifying a starting and ending port numbers with the **range** keyword. • Destination or source port-based filtering is enabled by configuring the **dst** or **src** keywords. • The example configures a master controller to learn prefixes from a database during each monitoring period. The database traffic is identified by a range of port numbers.

Step 13	**end**
	Example:
	`Router(config-oer-mc)# end`
		Exits OER Top Talker and Top Delay learning configuration mode, and returns to privileged EXEC mode.

What to Do Next

This section shows how to configure automatic prefix-based traffic class learning using protocol or port number. To configure specific prefix-based traffic classes using protocol or port numbers for OER monitoring and optimization, see the "Manually Selecting Traffic Classes Using Prefix, Protocol, Port, and DSCP Value" section on page 33.

Specifying the Flow Keys for Automatic Learning of Application Traffic Classes

Perform this task at the master controller to define the application traffic flow fields that OER can use to automatically learn traffic classes to be entered in the MTC list. In Cisco IOS Release 12.4(9)T, 12.2(33)SRB, and later releases, traffic class commands were introduced to help define the application traffic classes. The traffic class commands can be used in the following situations:

- You can use the filter and aggregation traffic class commands with the traffic class keys. Traffic class keys are specified, but they will be used only if the traffic class aggregation access list does not have any matches. In this situation, some knowledge of the prefixes that OER will learn is presumed.
- You can also use this task without the traffic class commands that use the filter and aggregation access lists, if you do not want to filter or aggregate any traffic classes. In this situation, no knowledge of the prefixes is presumed and only the traffic class command that specifies the keys is used.

In Cisco IOS Release 12.4(9)T and 12.2(33)SRB the ability to learn traffic using protocol, port number, and DSCP value (in addition to prefix) was introduced. Specifying the protocol, ports, and DSCP value allows application traffic to be identified in more detail. In this task, only traffic class keys are specified for voice traffic. The voice application traffic is identified by the UDP protocol, a DSCP value of ef, and port numbers in the range from 3000 to 4000. The master controller is also configured to learn the top prefixes based on highest outbound throughput for the specified traffic and the resulting traffic classes are added to the OER application database to be passively and actively monitored.

To display information about the traffic classes learned by OER use the"Displaying Application Traffic Flow Information on a Border Router" section on page 27.

Prerequisites

This task requires the master controller and border routers to be running Cisco IOS Release 12.4(9)T, 12.2(33)SRB, or later releases.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **oer master**
4. **learn**
5. **aggregation-type** {**bgp** | **non-bgp** | **prefix-length** *prefix-mask*}
6. **throughput**
7. **monitor-period** *minutes*
8. **periodic-interval** *minutes*
9. **prefixes** *number*

10. **traffic-class keys** [[**default**] | [**sport**] [**dport**] [**dscp**] [**protocol**]]
11. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		- Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`oer master`**
	Example:
	`Router(config)# oer master`
		Enters OER master controller configuration mode to configure a Cisco router as a master controller and to configure master controller policy and timer settings.

Step 4	**`learn`**
	Example:
	`Router(config-oer-mc)# learn`
		Enters OER Top Talker and Top Delay learning configuration mode to configure prefix learning policies and timers.

Step 5	**`aggregation-type`** `{`**`bgp`** `|` **`non-bgp`** `|` **`prefix-length`**`}` *`prefix-mask`*
	Example:
	`Router(config-oer-mc-learn)# aggregation-type prefix-length 24`
		(Optional) Configures a master controller to aggregate learned prefixes based on traffic flow type.
		- The **bgp** keyword configures prefix aggregation based on entries in the BGP routing table. This keyword is used if BGP peering is enabled in the network.
		- The **non-bgp** keyword configures learned prefix aggregation based on static routes. Entries in the BGP routing table are ignored when this keyword is entered.
		- The **prefix-length** keyword configures aggregation based on the specified prefix length. The range of values that can be configured for this argument is a prefix mask from 1 to 32.
		- If this command is not specified, the default aggregation is performed based on a /24 prefix length.
		- The example configures prefix length aggregation.

Step 6	**`throughput`**
	Example:
	`Router(config-oer-mc-learn)# throughput`
		Configures the master controller to learn the top prefixes based on the highest outbound throughput.
		- When this command is enabled, the master controller will learn the top prefixes across all border routers according to the highest outbound throughput.
		- The example configures a master controller to learn the top prefixes based on highest outbound throughput.

	Command or Action	Purpose
Step 7	**monitor-period** *minutes* **Example:** `Router(config-oer-mc-learn)# monitor-period 10`	(Optional) Sets the time period that an OER master controller learns traffic flows. • The default learning period is 5 minutes. • The length of time between monitoring periods is configured with the **periodic-interval** command. • The number of prefixes that are learned is configured with the **prefixes** command. • The example sets the length of each monitoring period to 10 minutes.
Step 8	**periodic-interval** *minutes* **Example:** `Router(config-oer-mc-learn)# periodic-interval 20`	(Optional) Sets the time interval between prefix learning periods. • By default, the interval between prefix learning periods is 120 minutes. • The example sets the time interval between monitoring periods to 20 minutes.
Step 9	**prefixes** *number* **Example:** `Router(config-oer-mc-learn)# prefixes 200`	(Optional) Sets the number of prefixes that the master controller will learn during the monitoring period. • By default, the top 100 traffic flows are learned. • The example configures a master controller to learn 200 prefixes during each monitoring period.
Step 10	**traffic-class keys** [[**default**] | [**sport**] [**dport**] [**dscp**] [**protocol**]] **Example:** `Router(config-oer-mc-learn)# traffic-class keys dport dscp protocol`	Specifies a key list used by the border router to aggregate the traffic flows into the learn aggregation cache. • Traffic class keys are used when there is no traffic class aggregation access list or if the traffic class aggregation access list does not have any matches. • The example specifies a key list of destination port, dscp value, and protocol.

Step 11	**end**
	Example:
	`Router(config-oer-mc-learn)# end`
		Exits OER Top Talker and Top Delay learning configuration mode, and returns to privileged EXEC mode.

Creating an Access List to Specify a Filter for Automatically Learned Application Traffic

Perform this task at the master controller to create an access list to filter specific application traffic for OER monitoring. In Cisco IOS Release 12.4(9)T and 12.2(33)SRB the ability to learn traffic using protocol, port number, and DSCP value (in addition to prefix) was introduced. Specifying the protocol, ports, and DSCP value allows application traffic to be identified in more detail.

In the Specifying the Flow Keys for Automatic Learning of Application Traffic Classes task, traffic keys were used to identify application traffic because no knowledge of any of the prefixes was assumed. If you know some prefixes that you want to exclude, then you can use this task to create an access list and filter out unwanted traffic. In this example for Voice traffic, the access list, VOICE_FILTER_LIST, configures OER to identify all UDP traffic from any source to a destination prefix of 10.1.0.0/16 with a DSCP value of ef that represents voice traffic. The access list is applied using a traffic class command that filters out unwanted traffic. The master controller is also configured to learn the top prefixes based on highest outbound throughput for the filtered traffic and the resulting traffic classes are added to the OER application database to be passively and actively monitored.

To display information about the traffic classes learned by OER use the"Displaying Application Traffic Flow Information on a Border Router" section on page 27.

Prerequisites

This task requires the master controller and border routers to be running Cisco IOS Release 12.4(9)T, 12.2(33)SRB, or later releases.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list** {**standard** | **extended**} *access-list-name*
4. [*sequence-number*] **permit udp** *source source-wildcard* [*operator* [*port*]] *destination destination-wildcard* [*operator* [*port*]] [**dscp** *dscp-value*]
5. **exit**
6. **oer master**
7. **learn**
8. **aggregation-type** {**bgp** | **non-bgp** | **prefix-length** *prefix-mask*}
9. **throughput**
10. **monitor-period** *minutes*
11. **periodic-interval** *minutes*
12. **prefixes** *number*
13. **traffic-class filter access-list** *access-list-name*
14. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip access-list** {**standard** | **extended**} *access-list-name* **Example:** `Router(config)# ip access-list extended VOICE_FILTER_LIST`	Defines an IP access list by name. • OER supports only named access lists. • The example creates an extended IP access list named VOICE_FILTER_LIST.
Step 4	[*sequence-number*] **permit udp** *source source-wildcard* [*operator* [*port*]] *destination destination-wildcard* [*operator* [*port*]] [**dscp** *dscp-value*] **Example:** `Router(config-ext-nacl)# permit udp any 10.1.0.0 0.0.255.255 dscp ef`	Sets conditions to allow a packet to pass a named IP access list. • The example is configured to identify all UDP traffic from any source to a destination prefix of 10.1.0.0/16 where the DSCP bit is set to ef. This specific UDP traffic is to be optimized. **Note** Only the syntax applicable to this task is shown. For more details, see the *Cisco IOS IP Application Services Command Reference*.
Step 5	**exit** **Example:** `Router(config-ext-nacl)# exit`	(Optional) Exits extended access list configuration mode and returns to global configuration mode.
Step 6	**oer master** **Example:** `Router(config)# oer master`	Enters OER master controller configuration mode to configure a Cisco router as a master controller and to configure master controller policy and timer settings.
Step 7	**learn** **Example:** `Router(config-oer-mc)# learn`	Enters OER Top Talker and Top Delay learning configuration mode to configure prefix learning policies and timers.
Step 8	**aggregation-type** {**bgp** | **non-bgp** | **prefix-length**} *prefix-mask* **Example:** `Router(config-oer-mc-learn)# aggregation-type prefix-length 24`	(Optional) Configures a master controller to aggregate learned prefixes based on traffic flow type. • The **bgp** keyword configures prefix aggregation based on entries in the BGP routing table. This keyword is used if BGP peering is enabled in the network. • The **non-bgp** keyword configures learned prefix aggregation based on static routes. Entries in the BGP routing table are ignored when this keyword is entered. • The **prefix-length** keyword configures aggregation based on the specified prefix length. The range of values that can be configured for this argument is a prefix mask from 1 to 32. • If this command is not specified, the default aggregation is performed based on a /24 prefix length. • The example configures prefix length aggregation.

	Command or Action	Purpose

Step 9	**`throughput`**
	Example:
	`Router(config-oer-mc-learn)# throughput`
		Configures the master controller to learn the top prefixes based on the highest outbound throughput.
		• When this command is enabled, the master controller will learn the top prefixes across all border routers according to the highest outbound throughput.
		• The example configures a master controller to learn the top prefixes based on highest outbound throughput.

Step 10	**`monitor-period`** *`minutes`*
	Example:
	`Router(config-oer-mc-learn)# monitor-period 10`
		(Optional) Sets the time period that an OER master controller learns traffic flows.
		• The default learning period is 5 minutes.
		• The length of time between monitoring periods is configured with the **periodic-interval** command.
		• The number of prefixes that are learned is configured with the **prefixes** command.
		• The example sets the length of each monitoring period to 10 minutes.

Step 11	**`periodic-interval`** *`minutes`*
	Example:
	`Router(config-oer-mc-learn)# periodic-interval 20`
		(Optional) Sets the time interval between prefix learning periods.
		• By default, the interval between prefix learning periods is 120 minutes.
		• The example sets the time interval between monitoring periods to 20 minutes.

Step 12	**`prefixes`** *`number`*
	Example:
	`Router(config-oer-mc-learn)# prefixes 200`
		(Optional) Sets the number of prefixes that the master controller will learn during the monitoring period.
		• By default, the top 100 traffic flows are learned.
		• The example configures a master controller to learn 200 prefixes during each monitoring period.

Step 13	**`traffic-class filter access-list`** *`access-list-name`*
	Example:
	`Router(config-oer-mc-learn)# traffic-class filter access-list VOICE_FILTER_LIST`
		Supports filtering of traffic classes during OER passive monitoring by using an extended access list.
		• The example configures learned prefixes to be filtered using the access list named VOICE_FILTER_LIST that was created in Step 3 of this task.

Step 14	**`end`**
	Example:
	`Router(config-oer-mc-learn)# end`
		Exits OER Top Talker and Top Delay learning configuration mode, and returns to privileged EXEC mode.

Creating an Access List to Specify Aggregation Criteria for Automatically Learned Application Traffic

Perform this task at the master controller to create an access list to aggregate learned application traffic for OER monitoring. In Cisco IOS Release 12.4(9)T and 12.2(33)SRB the ability to learn traffic using protocol, port number, and DSCP value (in addition to prefix) was introduced. Specifying the protocol, ports, and DSCP value allows application traffic to be identified in more detail.

In the Creating an Access List to Specify a Filter for Automatically Learned Application Traffic task, the application traffic was filtered to profile traffic for a specific destination prefix, but in this task, the application traffic is being aggregated for a range of destination ports. In this example, the access list, VOICE_AGG_LIST is configured to aggregate traffic with a destination port in the range from 3000 to 4000 and with a DSCP value of ef. This UDP traffic represents voice traffic and OER will create traffic classes based on the specified port number range and DSCP value. In this task, the master controller is also configured to learn the top prefixes based on highest outbound throughput for the aggregated traffic and the resulting traffic classes are added to the OER application database to be passively and actively monitored.

The last step in this task is an optional step to review the configuration on the OER master controller. To display more information about the traffic classes learned by OER use the“Displaying Application Traffic Flow Information on a Border Router” section on page 27.

Prerequisites

This task requires the master controller and border routers to be running Cisco IOS Release 12.4(9)T, 12.2(33)SRB, or later releases.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list** {**standard** | **extended**} *access-list-name*
4. [*sequence-number*] **permit udp** *source source-wildcard* [*operator* [*port*]] *destination destination-wildcard* [*operator* [*port*]] [**dscp** *dscp-value*]
5. **exit**
6. **oer master**
7. **learn**
8. **aggregation-type** {**bgp** | **non-bgp** | **prefix-length** *prefix-mask*}
9. **throughput**
10. **monitor-period** *minutes*
11. **periodic-interval** *minutes*
12. **prefixes** *number*
13. **traffic-class aggregate access-list** *access-list-name*
14. **end**
15. **show oer master**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		- Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`ip access-list`** {**`standard`** | **`extended`**} *`access-list-name`*
	Example:
	`Router(config)# ip access-list extended VOICE_AGG_LIST`
		Defines an IP access list by name.
		- OER supports only named access lists.
		- The example creates an extended IP access list named VOICE_AGG_LIST.

Step 4	[*`sequence-number`*] **`permit udp`** *`source source-wildcard`* [*`operator`* [*`port`*]] *`destination destination-wildcard`* [*`operator`* [*`port`*]] [**`dscp`** *`dscp-value`*] **Example:** `Router(config-ext-nacl)# permit udp any any range 3000 4000 dscp ef`	Sets conditions to allow a packet to pass a named IP access list. - The example is configured to identify all UDP traffic ranging from a destination port number of 3000 to 4000 from any source where the DSCP bit is set to ef. This specific UDP traffic is to be optimized. **Note** Only the syntax applicable to this task is shown. For more details, see the *Cisco IOS IP Application Services Command Reference*.

Step 5	**`exit`**
	Example:
	`Router(config-ext-nacl)# exit`
		(Optional) Exits extended access list configuration mode and returns to global configuration mode.

Step 6	**`oer master`**
	Example:
	`Router(config)# oer master`
		Enters OER master controller configuration mode to configure a Cisco router as a master controller and to configure master controller policy and timer settings.

Step 7	**`learn`**
	Example:
	`Router(config-oer-mc)# learn`
		Enters OER Top Talker and Top Delay learning configuration mode to configure prefix learning policies and timers.

	Command or Action	Purpose
Step 8	**aggregation-type** {**bgp** | **non-bgp** | **prefix-length**} *prefix-mask* **Example:** `Router(config-oer-mc-learn)# aggregation-type prefix-length 24`	(Optional) Configures a master controller to aggregate learned prefixes based on traffic flow type. • The **bgp** keyword configures prefix aggregation based on entries in the BGP routing table. This keyword is used if BGP peering is enabled in the network. • The **non-bgp** keyword configures learned prefix aggregation based on static routes. Entries in the BGP routing table are ignored when this keyword is entered. • The **prefix-length** keyword configures aggregation based on the specified prefix length. The range of values that can be configured for this argument is a prefix mask from 1 to 32. • If this command is not specified, the default aggregation is performed based on a /24 prefix length. • The example configures prefix length aggregation.
Step 9	**throughput** **Example:** `Router(config-oer-mc-learn)# throughput`	Configures the master controller to learn the top prefixes based on the highest outbound throughput. • When this command is enabled, the master controller will learn the top prefixes across all border routers according to the highest outbound throughput. • The example configures a master controller to learn the top prefixes based on highest outbound throughput.
Step 10	**monitor-period** *minutes* **Example:** `Router(config-oer-mc-learn)# monitor-period 10`	(Optional) Sets the time period that an OER master controller learns traffic flows. • The default learning period is 5 minutes. • The length of time between monitoring periods is configured with the **periodic-interval** command. • The number of prefixes that are learned is configured with the **prefixes** command. • The example sets the length of each monitoring period to 10 minutes.
Step 11	**periodic-interval** *minutes* **Example:** `Router(config-oer-mc-learn)# periodic-interval 20`	(Optional) Sets the time interval between prefix learning periods. • By default, the interval between prefix learning periods is 120 minutes. • The example sets the time interval between monitoring periods to 20 minutes.
Step 12	**prefixes** *number* **Example:** `Router(config-oer-mc-learn)# prefixes 200`	(Optional) Sets the number of prefixes that the master controller will learn during the monitoring period. • By default, the top 100 traffic flows are learned. • The example configures a master controller to learn 200 prefixes during each monitoring period.

	Command or Action	Purpose

Step 13	**`traffic-class aggregate access-list`** *`access-list-name`*
	Example:
	`Router(config-oer-mc-learn)# traffic-class aggregate access-list VOICE_AGG_LIST`
		Supports aggregation of traffic classes during OER passive monitoring by using an extended access list.
		• The example configures learned prefixes to be aggregated using the access list named VOICE_AGG_LIST that was created in Step 3 of this task.

Step 14	**`end`**
	Example:
	`Router(config-oer-mc-learn)# end`
		Exits OER Top Talker and Top Delay learning configuration mode, and returns to privileged EXEC mode.

Step 15	**`show oer master`**
	Example:
	`Router# show oer master`
		(Optional) Displays information about the status of the OER-managed network; the output includes information about the master controller, the border routers, OER managed interfaces, and default and user-defined policy settings.

Examples

The following example output for the **show oer master** command displays the additional configuration for the traffic class aggregation, filters, and key list under the Learn Settings section.

```
Router# show oer master

OER state: ENABLED and ACTIVE
 Conn Status: SUCCESS, PORT: 7777
  Version: 2.0
  Number of Border routers: 2
  Number of Exits: 2
  Number of monitored prefixes: 0 (max 5000)
  Max prefixes: total 5000 learn 2500
  Prefix count: total 0, learn 0, cfg 0

Border           Status   UP/DOWN             AuthFail  Version
1.1.1.2          ACTIVE   UP       00:18:57          0  2.0
1.1.1.1          ACTIVE   UP       00:18:58          0  2.0

Global Settings:
  max-range-utilization percent 20 recv 20
  mode route metric bgp local-pref 5000
  mode route metric static tag 5000
  trace probe delay 1000
  logging

Default Policy Settings:
  backoff 180 200 180
  delay relative 50
  holddown 300
  periodic 0
  probe frequency 56
  mode route control
  mode monitor active
 mode select-exit good
  loss relative 10
  jitter threshold 20
  mos threshold 3.60 percent 30
  unreachable relative 50
  resolve delay priority 11 variance 20
  resolve utilization priority 12 variance 20
 *tag 0

Learn Settings:
  current state : STARTED
  time remaining in current state : 70 seconds
  throughput
  no delay
  no inside bgp
  traffic-class filter access-list voice-filter-acl <----
  traffic-class aggregate access-list voice-agg-acl <----
  traffic-class keys protocol dscp dport  <----
  no protocol
  monitor-period 2
  periodic-interval 1
  aggregation-type prefix-length 24
  prefixes 10
  expire after time 720
```

Displaying Application Traffic Flow Information on a Border Router

Perform this task to display application traffic flow information. These commands are entered on a border router through which the application traffic is flowing. The commands can be entered in any order. Keywords in Step 2 and Step 4 require the border router to be running Cisco IOS Release 12.4(9)T, 12.2(33)SRB, 12.2(33)SXH, or later releases.

Prerequisites

This task requires the master controller and border routers to be running Cisco IOS Release 12.4(9)T, 12.2(33)SRB, or later releases.

SUMMARY STEPS

1. **enable**
2. **show oer border passive learn**
3. **show ip cache verbose flow**
4. **show oer border passive cache** {**learned** | **prefix**} [**applications**]

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

```
Router> enable
```

Step 2 **show oer border passive learn**

This command is used to display traffic class filter and aggregation ACL information. The following example displays the voice application filter, aggregation, and keys information configured in the first three tasks under the "Specifying the Flow Keys for Automatic Learning of Application Traffic Classes" task.

```
Router# show oer border passive learn

OER Border Learn Configuration :
    State is enabled
    Measurement type: throughput, Duration: 2 min
    Aggregation type: prefix-length, Prefix length: 24
    No port protocol config

 Traffic Class Filter List:
   List: SrcPrefix       SrcMask DstPrefix       DstMask
         Prot  DSCP  sport_opr sport_range    dport_opr dport_range     Grant
      1: 0.0.0.0         0       10.1.0.0        16
         17      ef  0         [1, 65535]      0        [1, 65535]      Permit

 Traffic Class Aggregate List:
   List: Prot  DSCP  sport_opr sport_range    dport_opr dport_range     Grant
      1: 17      ef  0         [1, 65535]      7        [3000, 4000]    Permit

 Keys:  protocol dscp DstPort
```

Step 3 **show ip cache verbose flow**

This is a NetFlow command that is used to display all the flows (including applications) currently active on the border router. The following example displays traffic flow statistics by protocol, source address, and destination:

```
Router# show ip cache verbose flow
IP packet size distribution (203337 total packets):
   1-32   64   96  128  160  192  224  256  288  320  352  384  416  448  480
   .397 .602 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000

    512  544  576 1024 1536 2048 2560 3072 3584 4096 4608
   .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000

IP Flow Switching Cache, 278544 bytes
  5 active, 4091 inactive, 310 added
  47486 ager polls, 0 flow alloc failures
  Active flows timeout in 30 minutes
  Inactive flows timeout in 15 seconds
IP Sub Flow Cache, 25800 bytes
  13 active, 1011 inactive, 355 added, 310 added to flow
  0 alloc failures, 0 force free
  1 chunk, 1 chunk added
  last clearing of statistics never

Protocol         Total    Flows   Packets Bytes  Packets Active(Sec) Idle(Sec)
--------         Flows     /Sec     /Flow  /Pkt     /Sec       /Flow   /Flow
TCP-other           14      0.0      7370    40      9.7    1556.8     3.4
UDP-other            9      0.0      7579    28      6.4    1601.0     3.5
ICMP               282      0.0         1    64      0.0       0.0    15.6
Total:             305      0.0       562    35     16.3     118.7    14.7

SrcIf          SrcIPaddress    DstIf          DstIPaddress    Pr TOS Flgs  Pkts
Port Msk AS                    Port Msk AS    NextHop              B/Pk  Active
Et8/0          172.20.1.1      Et0/0          10.1.3.1        11 B8  10     6334
07D0 /0  0                     0DAC /0  0     10.40.40.2           28  1337.8
Et8/0          172.20.1.1      Et0/0          10.2.2.1        06 00  00     6338
07D0 /0  0                     0DAC /0  0     10.40.40.2           40  1338.6
Et8/0          172.20.1.1      Et0/0          10.1.3.1        06 00  00     6333
07D0 /0  0                     0DAC /0  0     10.40.40.2           40  1337.6
Et8/0          172.20.1.1      Et0/0          10.1.1.1        06 00  00     6334
07D0 /0  0                     1964 /0  0     10.40.40.2           40  1337.8
Et8/0          172.20.1.1      Et0/0          10.1.1.1        11 B8  10     6339
07D0 /0  0                     0E10 /0  0     10.40.40.2           28  1338.8
```

```
Total number of prefixes 2
```

Step 4 **show oer border passive cache** {**learned** | **prefix**} [**applications**]

This command is used to display real-time prefix information collected from the border router through NetFlow passive monitoring. Using the **learned** and **applications** keywords you can display information about learned applications. In the output you can see that only application traffic classes matching the traffic class keys, filter, and aggregation criteria set in the first three tasks under the "Specifying the Flow Keys for Automatic Learning of Application Traffic Classes" task are saved in the learn cache.

```
Router# show oer border passive cache learned applications

OER Learn Cache:
    State is enabled
    Measurement type: throughput, Duration: 2 min
    Aggregation type: prefix-length, Prefix length: 24
    4096 oer-flows per chunk,
    8 chunks allocated, 32 max chunks,
    5 allocated records, 32763 free records, 4588032 bytes allocated

Prefix         Mask     Pkts  B/Pk  Delay Samples   Active
Prot  Dscp  SrcPort          DstPort
Host1          Host2          Host3          Host4          Host5
dport1         dport2         dport3         dport4         dport5
10.1.3.0        /24     873     28      0      0      13.3
17      ef [1, 65535]       [3000, 4000]
10.1.3.1       0.0.0.0        0.0.0.0        0.0.0.0        0.0.0.0
3500            0               0               0               0
10.1.1.0        /24    7674     28      0      0      13.4
17      ef [1, 65535]       [3000, 4000]
10.1.1.1       0.0.0.0        0.0.0.0        0.0.0.0        0.0.0.0
3600            0               0               0               0
```

What To Do Next

More information about monitoring and measuring traffic flow information for applications is documented in the "Measuring the Traffic Class Performance and Link Utilization Using OER" module.

Manually Selecting Prefixes for OER Monitoring

Perform this task to manually select prefixes for monitoring. An IP prefix list is created to define the prefix or prefix range. The prefix list is then imported into the central policy database by configuring a match clause in an OER map. For details about using IP prefix lists with OER, see the "Prefix Traffic Class Configuration Using OER" section on page 6.

OER Map Operation for the OER Profile Phase

An OER map may appear to be similar to a route map but there are significant differences. An OER map is configured to select an IP prefix list using a match clause. The OER map is configured with a sequence number like a route map, and the OER map with the lowest sequence number is evaluated first. The operation of an OER map differs from a route map at this point. There are two important distinctions:

- Only a single match clause may be configured for each sequence. An error message will be displayed on the console if you attempt to configure multiple match clauses for a single OER map sequence.
- An OER map is not configured with permit or deny statements. However, a permit or deny sequence can be configured for an IP traffic flow by configuring a permit or deny statement in an IP prefix list and then applying the prefix list to the OER map.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network/length* | **permit** *network/length*} [**le** *le-value*]
4. **oer-map** *map-name sequence-number*
5. **match ip address prefix-list** *name*
6. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`ip prefix-list`** *`list-name`* `[`**`seq`** *`seq-value`*`] {`**`deny`** *`network/length`* `|` **`permit`** *`network/length`*`} [`**`le`** *`le-value`*`]`
	Example:
	`Router(config)# ip prefix-list PREFIXES seq 20 permit 10.1.5.0/24`
		Creates a prefix list to manually select prefixes for monitoring.
		• A master controller can monitor and control an exact prefix of any length including the default route. The master controller acts only on the configured prefix.
		• A master controller can monitor and control an inclusive prefix using the le 32 option. The master controller acts on the configured prefix and forces any more specific prefixes in the RIB to use the same exit.
		Note This option should be applied carefully. It is not needed in typical deployments.
		• The example creates an IP prefix list for OER to monitor and control the exact prefix, 10.1.5.0/24

Step 4	**`oer-map map-name`** *`sequence-number`*
	Example:
	`Router(config)# oer-map IMPORT 10`
		Enters OER map configuration mode to create or configure an OER map.
		• *Only a single match clause can be configured for each OER map sequence.*
		• The example creates an OER map named IMPORT.

	Command or Action	Purpose
Step 5	**match ip address prefix-list** *name* **Example:** `Router(config-oer-map)# match ip address prefix-list PREFIXES`	Creates a prefix list match clause entry in an OER map to apply OER policies. • This command supports IP prefix lists only. • The example configures the prefix list PREFIXES.

Step 6	**end**
	Example:
	`Router(config-oer-map)# end`
		Exits OER map configuration mode and returns to privileged EXEC mode.

What to Do Next

This section shows how to manually configure prefix learning. To configure automatic prefix learning, see the "Configuring OER to Automatically Learn Prefix-Based Traffic Classes" section on page 8.

Manually Selecting Inside Prefixes for OER Monitoring

In Cisco IOS Release 12.4(9)T, 12.2(33)SRB, and later releases, the OER BGP inbound optimization feature introduced the ability to manually select inside prefixes to support best entrance selection for traffic that originates from prefixes outside an autonomous system destined for prefixes inside the autonomous system. Perform this task to manually select inside prefixes for OER monitoring by creating an IP prefix list to define the inside prefix or prefix range. The prefix list is then imported into the MTC list by configuring a match clause in an OER map. For details about using IP prefix lists with OER, see the "Prefix Traffic Class Configuration Using OER" section on page 6.

OER Inside Prefixes

An OER inside prefix is defined as a public IP prefix assigned to a company. An OER outside prefix is defined as a public IP prefix assigned outside the company. Companies advertise the inside prefixes over the Internet using an Internet service provider (ISP) and receive advertisements for outside prefixes from an ISP.

OER Map Operation for Inside Prefixes

The operation of an OER map is similar to the operation of a route-map. An OER map is configured to select an IP prefix list or OER learn policy using a match clause and then to apply OER policy configurations using a set clause. The OER map is configured with a sequence number like a route-map, and the OER map with the lowest sequence number is evaluated first. In Cisco IOS Release 12.4(9)T and 12.2(33)SRB, the **inside** keyword that identifies inside prefixes was added to the **match ip address** (OER) command.

Prerequisites

This task requires the master controller and border routers to be running Cisco IOS Release 12.4(9)T, 12.2(33)SRB, or later releases.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network/length* | **permit** *network/length*} [**le** *le-value*]
4. **oer-map** *map-name sequence-number*
5. **match ip address prefix-list** *name* [**inside**]
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network/length* | **permit** *network/length*} [**le** *le-value*] **Example:** `Router(config)# ip prefix-list INSIDE_PREFIXES seq 20 permit 192.168.1.0/24`	Creates a prefix list to manually select prefixes for monitoring. • A master controller can monitor and control an exact prefix of any length including the default route. The master controller acts only on the configured prefix. • A master controller can monitor and control an inclusive prefix using the le 32 option. The master controller acts on the configured prefix and forces any more specific prefixes in the RIB to use the same exit. **Note** This option should be applied carefully. It is not needed in typical deployments. • The example creates an IP prefix list for OER to monitor and control the exact prefix, 192.168.1.0/24
Step 4	**oer-map map-name** *sequence-number* **Example:** `Router(config)# oer-map INSIDE_MAP 10`	Enters OER map configuration mode to create or configure an OER map. • *OER map operation is similar to that of route maps.* • *Only a single match clause can be configured for each OER map sequence.* • Common and deny sequences should be applied to lowest OER map sequence for best performance. • The example creates an OER map named INSIDE_MAP.

	Command or Action	Purpose
Step 5	**match ip address prefix-list** *name* [**inside**] **Example:** `Router(config-oer-map)# match ip address prefix-list INSIDE_PREFIXES inside`	Creates a prefix list match clause entry in an OER map to apply OER policies. • This command supports IP prefix lists only. • Use the **inside** keyword to identify inside prefixes. • The example creates a match clause to use the prefix list INSIDE_PREFIXES to specify that inside prefixes must be matched.

Step 6	**end**
	Example:
	`Router(config-oer-map)# end`
		Exits OER map configuration mode and returns to privileged EXEC mode.

What to Do Next

This section shows how to configure specific inside prefixes for OER monitoring and optimization. To configure automatic prefix learning for inside prefixes, see the "Configuring OER to Automatically Learn Traffic Classes Using Inside Prefixes" section on page 11.

Manually Selecting Traffic Classes Using Prefix, Protocol, Port, and DSCP Value

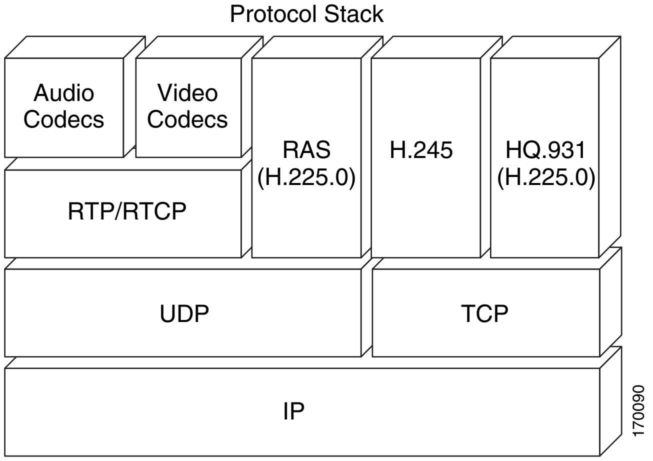
Perform this task to manually select traffic classes using prefixes, protocols, port numbers, and DSCP value for OER monitoring. An IP access list is created to define the parameters to identify the traffic classes. The access list can then be imported into the MTC list by configuring a match clause in an OER map.

This example task uses an access list to identify voice traffic. Before voice traffic can be optimized, it must be identified. In this task, the voice traffic that is to be optimized is identified by a protocol of UDP, a range of source and destination port numbers from 16384 to 32767, a destination prefix of 10.20.20.0/24, and a DSCP value of ef.

IP Protocol Stack for Voice

Voice traffic uses a variety of protocols and streams on the underlying IP network. Figure 3 is a representation of the protocol options available for carrying voice traffic over IP. Most signaling traffic for voice is carried over TCP. Most voice calls are carried over User Datagram Protocol (UDP) and Real-Time Protocol (RTP). You can configure your voice devices to use a specific range of destination port numbers over UDP to carry voice call traffic.

Figure 3 Protocol Stack Options Available for Voice Traffic



Prerequisites

This task requires the master controller and border routers to be running Cisco IOS Release 12.4(9)T, 12.2(33)SRB, or later releases.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access list** {**standard** | **extended**} *access-list-name*
4. [*sequence-number*] **permit udp** *source source-wildcard* [*operator* [*port*]] *destination destination-wildcard* [*operator* [*port*]] [**dscp** *dscp-value*]
5. **exit**
6. **oer-map** *map-name sequence-number*
7. **match ip address** {**access-list** *access-list-name* | **prefix-list** *prefix-list-name*}
8. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip access-list** {**standard** | **extended**} *access-list-name* **Example:** `Router(config)# ip access-list extended VOICE_ACCESS_LIST`	Defines an IP access list by name. • OER supports only named access lists. • The example creates an extended IP access list named VOICE_ACCESS_LIST.
Step 4	[*sequence-number*] **permit udp** *source source-wildcard* [*operator* [*port*]] *destination destination-wildcard* [*operator* [*port*]] [**dscp** *dscp-value*] **Example:** `Router(config-ext-nacl)# permit udp any range 16384 32767 10.20.20.0 0.0.0.15 range 16384 32767 dscp ef`	Sets conditions to allow a packet to pass a named IP access list. • The example is configured to identify all UDP traffic with a source or destination port number in the range from 16384 to 32767 from any source prefix to a destination prefix of 10.20.20.0/24, and with a DSCP value of ef. This specific UDP traffic represents voice traffic. • Only the syntax applicable to this task is shown. For more details, see the *Cisco IOS IP Application Services Command Reference*, Release 12.4T
Step 5	**exit** **Example:** `Router(config-ext-nacl)# exit`	(Optional) Exits extended access list configuration mode and returns to global configuration mode.
Step 6	**oer-map** *map-name sequence-number* **Example:** `Router(config)# oer-map VOICE_MAP 10`	Enters OER map configuration mode to configure an OER map to apply policies to selected IP prefixes. • *Only one match clause can be configured for each OER map sequence.* • Permit sequences are first defined in an IP prefix list and then applied with the **match ip address** (OER) command in Step 7. • The example creates an OER map named VOICE_MAP.

Step 7	**match ip address** {**access-list** *access-list-name* | **prefix-list** *prefix-list-name*}
	Example:
	`Router(config-oer-map)# match ip address access-list VOICE_ACCESS_LIST`
		References an extended IP access list or IP prefix as match criteria in an OER map.
		• Only a single match clause can be configured for each OER map sequence.
		• The example configures the IP access list named VOICE_ACCESS_LIST as match criteria in an OER map.

Step 8	**end**
	Example:
	`Router(config-oer-map)# end`
		(Optional) Exits OER map configuration mode and returns to privileged EXEC mode.

What to Do Next

This section shows how to manually select traffic classes using prefixes, protocols, port numbers, and DSCP value for OER monitoring. To configure automatic learning of traffic classes using prefixes, protocols, port numbers, and DSCP values, see the "Specifying the Flow Keys for Automatic Learning of Application Traffic Classes" section on page 17.

Configuration Examples for Using OER to Profile the Traffic Classes

The examples in this section show how to configure automatic prefix learning and how to select specific prefixes for monitoring.

- Configuring OER to Automatically Learn Prefix-Based Traffic Classes: Example, page 36
- Configuring OER to Automatically Learn Traffic Classes Using Inside Prefixes: Example, page 36
- Configuring OER to Automatically Learn Traffic Classes Using Prefixes and Protocol or Port Numbers: Example, page 37
- Configuring OER to Automatically Learn Traffic Classes Using Protocol, Ports, and DSCP Value: Example, page 37
- Manually Selecting Prefixes for OER Monitoring: Example, page 38
- Manually Selecting Inside Prefixes for OER Monitoring: Example, page 38
- Manually Selecting Traffic Classes Using Prefix, Protocol, Port, and DSCP Value: Example, page 39

Configuring OER to Automatically Learn Prefix-Based Traffic Classes: Example

The following example, starting in global configuration mode, configures the master controller to automatically learn top prefixes based on the highest delay. The prefix monitoring period is set to 10 minutes. The number of prefixes that are monitored during each monitoring period is set to 500. The time interval between each monitoring period is set to 20 minutes.

```
Router(config)# oer master
Router(config-oer-master)# learn
Router(config-oer-master-learn)# delay
Router(config-oer-master-learn)# aggregation-type bgp
Router(config-oer-master-learn)# monitor-period 10
Router(config-oer-master-learn)# periodic-interval 20
Router(config-oer-master-learn)# prefixes 500
Router(config-oer-master-learn)# end
```

Configuring OER to Automatically Learn Traffic Classes Using Inside Prefixes: Example

The following example shows how to configure OER to automatically learn prefixes inside the network:

```
Router> enable
Router# configure terminal
Router(config)# oer master
Router(config-oer-mc)# learn
Router(config-oer-mc-learn)# inside bgp
Router(config-oer-mc-learn)# monitor-period 10
Router(config-oer-mc-learn)# periodic-interval 20
Router(config-oer-mc-learn)# prefixes 500
Router(config-oer-mc-learn)# end
```

Configuring OER to Automatically Learn Traffic Classes Using Prefixes and Protocol or Port Numbers: Example

The following example, starting in global configuration mode, learns traffic for SSH sessions that use 49152 as the destination port number in the IP packet header.

```
Router(config)# oer master
Router(config-oer-master)# learn
Router(config-oer-master-learn)# throughput
Router(config-oer-master-learn)# aggregation-type bgp
Router(config-oer-master-learn)# monitor-period 10
Router(config-oer-master-learn)# periodic-interval 20
Router(config-oer-master-learn)# protocol 22 port 49152 dst
Router(config-oer-master-learn)# end
```

Configuring OER to Automatically Learn Traffic Classes Using Protocol, Ports, and DSCP Value: Example

The following example, starting in global configuration mode, configures the master controller to automatically learn defined application traffic. Using a series of traffic class commands under OER learn configuration mode, only voice traffic with a DSCP bit set to ef, a protocol of UDP, and a destination port in the range of 3000 to 4000 is learned and added to the OER MTC list on the master controller.

The prefix monitoring period is set to 2 minutes. The number of prefixes that are monitored during each monitoring period is set to 10. The time interval between each monitoring period is set to 20 minutes.

```
Router(config)# ip access-list extended voice-filter-acl
Router(config-ext-nacl)# permit udp any 10.1.0.0 0.0.255.255 dscp ef
Router(config-ext-nacl)# exit
Router(config)# ip access-list extended voice-agg-acl
Router(config-ext-nacl)# permit udp any any range 3000 4000 dscp ef
Router(config-ext-nacl)# exit
Router(config)# oer master
Router(config-oer-master)# learn
Router(config-oer-master-learn)# aggregation-type prefix-length 24
Router(config-oer-master-learn)# throughput
Router(config-oer-master-learn)# monitor-period 2
Router(config-oer-master-learn)# periodic-interval 1
Router(config-oer-master-learn)# prefixes 10
Router(config-oer-master-learn)# traffic-class filter access-list voice-filter-acl
Router(config-oer-master-learn)# traffic-class aggregate access-list voice-agg-acl
Router(config-oer-master-learn)# traffic-class keys protocol dport dscp
Router(config-oer-master-learn)# end
```

More details about the OER network configuration for the example shown above can be seen in the running configuration file:

```
Router# show running-config

oer master
port 7777
```

```
logging
!
border 10.1.1.1 key-chain key1
 interface Serial12/0 external
 interface Ethernet8/0 internal
!
border 10.1.1.2 key-chain key2
 interface Ethernet0/0 external
 interface Ethernet8/0 internal
!
learn
 throughput
 periodic-interval 1
 monitor-period 2
 prefixes 10
 traffic-class filter access-list voice-filter-acl
 traffic-class aggregate access-list voice-agg-acl
 traffic-class keys protocol dscp dport
 backoff 180 200
 mode route control
 mode monitor active
!
active-probe echo 10.1.2.1
active-probe echo 10.1.1.1
active-probe echo 10.1.3.1
```

Manually Selecting Prefixes for OER Monitoring: Example

The following example, starting in global configuration mode, configures an OER map to exclude traffic from the 192.168.0.0/16 network and include traffic from the 10.5.5.0/24 network. Excluded prefixes are not imported into the MTC list.

```
Router(config)# ip prefix-list seq 10 EXCLUDE deny 192.168.0.0/16 le 32
Router(config)# ip prefix-list seq 10 IMPORT permit 10.5.5.0/24
Router(config)# oer-map PREFIXES 10
Router(config-oer-map)# match ip address prefix-list EXCLUDE
Router(config-oer-map)# exit
Router(config)# oer-map PREFIXES 20
Router(config-oer-map)# match ip address prefix-list IMPORT
Router(config-oer-map)# end
```

Manually Selecting Inside Prefixes for OER Monitoring: Example

The following example shows how to manually configure OER to learn prefixes inside the network using an OER map:

```
Router> enable
Router# configure terminal
Router(config)# ip prefix-list INSIDE_PREFIXES seq 20 permit 192.168.1.0/24
Router(config)# oer-map INSIDE_MAP 10
Router(config-oer-map)# match ip address prefix-list INSIDE_PREFIXES inside
Router(config-oer-map)# end
```

Manually Selecting Traffic Classes Using Prefix, Protocol, Port, and DSCP Value: Example

The following configuration is performed on an edge router which is both an OER master controller and a border router (for example, in a remote office network) to identify voice traffic using an extended named access list.

```
Router> enable
Router# configure terminal
Router(config)# ip access-list extended Voice_Traffic
Router(config-ext-nacl)# 10 permit udp any 10.1.0.0 0.0.255.255 range 16384 32767 dscp ef
Router(config-ext-nacl)# exit
Router(config)# oer-map Voice_MAP 10
Router(config-oer-map)# match ip address access-list Voice_Traffic
Router(config-oer-map)# end
```

Where To Go Next

This module covered the OER profile phase and it has assumed that you started with the "Cisco IOS Optimized Edge Routing Overview" and the "Setting Up OER Network Components" module. The profile phase is the first phase in the OER performance loop. To learn more about the other OER phases, read through the other modules in the following list:

- Measuring the Traffic Class Performance and Link Utilization Using OER
- Configuring and Applying OER Policies
- Using OER to Control Traffic Classes and Verify the Route Control Changes

After you understand the various OER phases, review the OER solutions modules that are listed under "Related Documents" section on page 39.

Additional References

The following sections provide references related to using OER to profile the traffic classes.

Related Documents

Related Topic	Document Title
Cisco OER technology overview	"Cisco IOS Optimized Edge Routing Overview" module
Concepts and configuration tasks required to set up OER network components	"Setting Up OER Network Components" module
OER solution module: voice traffic optimization using OER active probes.	"OER Voice Traffic Optimization Using Active Probes" module
OER solution module: configuring VPN IPsec/GRE tunnel interfaces as OER-managed exit links.	"Configuring VPN IPsec/GRE Tunnel Interfaces As OER-Managed Exit Links" module

Related Topic	Document Title
Cisco OER commands: complete command syntax, command mode, command history, defaults, usage guidelines and examples	*Cisco IOS Optimized Edge Routing Command Reference*
IP prefix list commands	*Cisco IOS IP Routing Protocols Command Reference*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Using OER to Profile the Traffic Classes

Table 1 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.3(11)T, 12.2(33)SRB, or a later release appear in the table.

For information on a feature in this technology that is not documented here, see the "Cisco IOS Optimized Edge Routing Features Roadmap."

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Using OER to Profile the Traffic Classes

Feature Name	Releases	Feature Configuration Information
Port and Protocol Based Prefix Learning	12.3(11)T 12.2(33)SRB	Port and protocol based prefix learning allows you to configure a master controller to learn prefixes based on the protocol type and TCP or UDP port number. The following sections provide information about this feature: • Prefix Traffic Class Learning Using OER, page 4 • Prefix Traffic Class Configuration Using OER, page 6 • Configuring OER to Automatically Learn Prefix-Based Traffic Classes Using Protocol or Port Number, page 13 • Manually Selecting Traffic Classes Using Prefix, Protocol, Port, and DSCP Value, page 33 • Configuring OER to Automatically Learn Traffic Classes Using Prefixes and Protocol or Port Numbers: Example, page 37 • Manually Selecting Traffic Classes Using Prefix, Protocol, Port, and DSCP Value: Example, page 39 The **protocol** command was introduced by this feature.
expire command[1]	12.3(14)T 12.2(33)SRB	The **expire** command is used to set an expiration period for learned prefixes. By default, the master controller removes inactive prefixes from the central policy database as memory is needed. This command allows you to refine this behavior by setting a time or session based limit. The time based limit is configured in minutes. The session based limit is configured for the number of monitor periods (or sessions).
OER Application-Aware Routing: PBR	12.4(2)T 12.2(33)SRB	The OER Application-Aware Routing: PBR feature introduces the capability to optimize IP traffic based on the type of application that is carried by the monitored prefix. Independent policy configuration is applied to the subset (application) of traffic. The following sections provide information about this feature: • Application Traffic Class Configuration Using OER, page 7 The following commands were introduced or modified by this feature: **debug oer border pbr**, **debug oer master prefix**, **match ip address (OER)**, **show oer master active-probes**, and **show oer master appl**.

Table 1 Feature Information for Using OER to Profile the Traffic Classes (continued)

Feature Name	Releases	Feature Configuration Information
OER BGP Inbound Optimization	12.4(9)T 12.2(33)SRB	OER BGP inbound optimization supports best entrance selection for traffic that originates from prefixes outside an autonomous system destined for prefixes inside the autonomous system. External BGP (eBGP) advertisements from an autonomous system to an Internet service provider (ISP) can influence the entrance path for traffic entering the network. OER uses eBGP advertisements to manipulate the best entrance selection. The following sections provide information about this feature: • Prefix Traffic Class Learning Using OER, page 4 • Configuring OER to Automatically Learn Traffic Classes Using Inside Prefixes, page 11 • Manually Selecting Inside Prefixes for OER Monitoring, page 31 • Configuring OER to Automatically Learn Traffic Classes Using Inside Prefixes: Example, page 36 • Manually Selecting Inside Prefixes for OER Monitoring: Example, page 38 The following commands were introduced or modified by this feature: **clear oer master prefix**, **downgrade bgp**, **inside bgp**, **match ip address (OER)**, **match oer learn**, **max range receive**, **maximum utilization receive**, **show oer master prefix**.

Table 1 Feature Information for Using OER to Profile the Traffic Classes (continued)

Feature Name	Releases	Feature Configuration Information
OER DSCP Monitoring	12.4(9)T 12.2(33)SRB	OER DSCP Monitoring introduced automatic learning of traffic classes based on protocol, port numbers, and DSCP value. Traffic classes can be defined by a combination of keys comprising of protocol, port numbers, and DSCP values, with the ability to filter out traffic that is not required, and the ability to aggregate the traffic in which you are interested. Layer 4 information such as protocol, port number, and DSCP information is now sent to the master controller database in addition to the Layer 3 prefix information. The new functionality allows OER to both actively and passively monitor application traffic. The following sections provide information about this feature: • Application Traffic Class Learning Using OER, page 4 • Application Traffic Class Configuration Using OER, page 7 • Specifying the Flow Keys for Automatic Learning of Application Traffic Classes, page 17 • Configuring OER to Automatically Learn Traffic Classes Using Protocol, Ports, and DSCP Value: Example, page 37 The following commands were introduced or modified by this feature: **show oer border passive applications**, **show oer border passive cache**, **show oer border passive learn**, **show oer master appl**, **traffic-class aggregation**, **traffic-class filter**, and **traffic-class keys**.

Table 1 Feature Information for Using OER to Profile the Traffic Classes (continued)

Feature Name	Releases	Feature Configuration Information
OER Border Router Only Functionality	12.2(33)SXH	In Cisco IOS Release 12.2(33)SXH support for using a Cisco Catalyst 6500 series switch as an OER border router was introduced. Only border router functionality is included in the Cisco IOS Release 12.2(33)SXH images; no master controller configuration is available. The master controller that communicates with the Cisco Catalyst 6500 series switch being used as a border router must be a router running Cisco IOS Release 12.4(6)T or a later release. The OER master controller software has been modified to handle the limited functionality supported by the Cisco Catalyst 6500 border routers. Using the Route Processor (RP), the Catalyst 6500 border routers can capture throughput statistics only for a traffic class compared to the delay, loss, unreachability, and throughput statistics collected by non-Catalyst 6500 border routers. A master controller automatically detects the limited capabilities of the Catalyst 6500 border routers and downgrades other border routers to capture only the throughput statistics for traffic classes. By ignoring other types of statistics, the master controller is presented with a uniform view of the border router functionality. The following sections provide information about this feature: • Restrictions for Using OER to Profile the Traffic Classes, page 2 • Displaying Application Traffic Flow Information on a Border Router, page 27 The following command was introduced or modified by this feature: **show oer border passive cache**.

1. This is a minor enhancement. Minor enhancements are not typically listed in Feature Navigator.

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

Measuring the Traffic Class Performance and Link Utilization Using OER

First Published: January 29, 2007
Last Updated: August 16, 2007

This module describes the Cisco IOS Optimized Edge Routing (OER) measure phase, which is the second step in the OER performance loop. In the measure phase, OER monitors the performance metrics of the traffic class entries that were identified during the OER profile phase. OER also monitors the link utilization in the measure phase. Monitoring is the act of measurement and comparison against a threshold to determine the occurrence of an out-of-policy (OOP) event. OER uses two types of measurement; active and passive monitoring.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Measuring the Traffic Class Performance and Link Utilization Using OER" section on page 48.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Measuring the Traffic Class Performance and Link Utilization Using OER

- Before implementing traffic class performance monitoring using OER, you need to understand and configure a basic OER-managed network. See the "Cisco IOS Optimized Edge Routing Overview" and "Setting Up OER Network Components" modules for more details.
- If you are following the OER performance loop we recommend that you understand and configure tasks in the "Using OER to Profile the Traffic Classes" module before attempting the tasks in this module.

Information About Measuring the Traffic Class Performance and Link Utilization Using OER

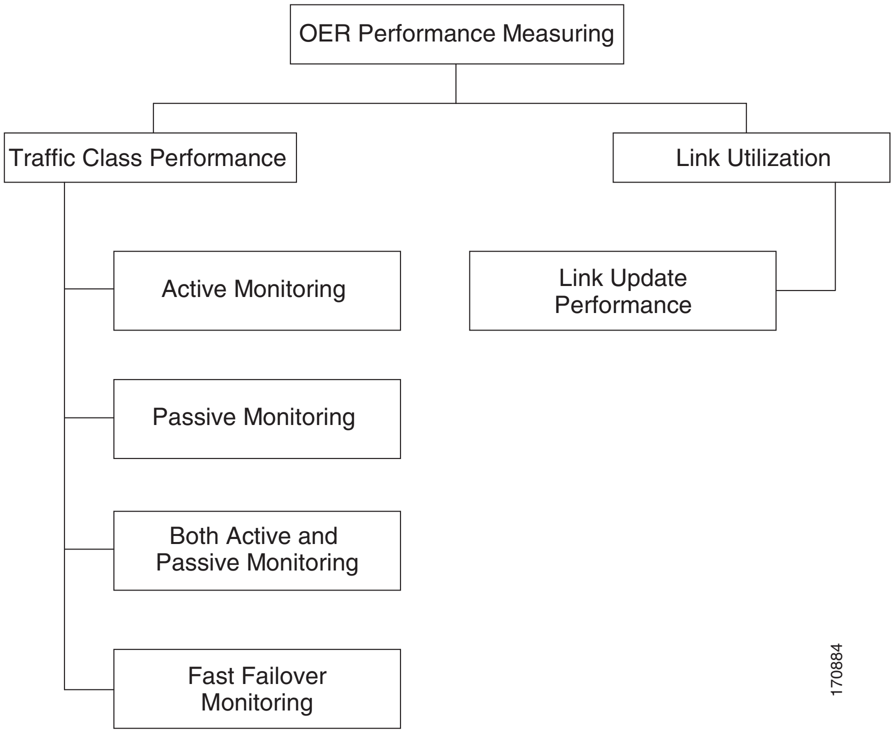
To configure traffic class performance monitoring on a master controller, you should understand the following concepts:

OER Measure Phase

The OER measure phase is the second step in the OER performance loop and it follows the OER profile phase where the traffic class entries fill the Monitored Traffic Class (MTC) list. The MTC list is now full of traffic class entries and OER must measure the performance metrics of these traffic class entries. Monitoring is defined here as the act of measurement performed periodically over a set interval of time where the measurements are compared against a threshold. OER measures the performance of traffic classes using active and passive monitoring techniques but it also measures, by default, the utilization of links. The master controller can be configured to monitor learned and configured traffic classes. The border routers collect passive monitoring and active monitoring statistics and then transmit this information to the master controller. The OER measure phase is complete when each traffic class entry in the MTC list has associated performance metric measurements.

The overall structure of the OER measure phase and its component parts can be seen in Figure 1.

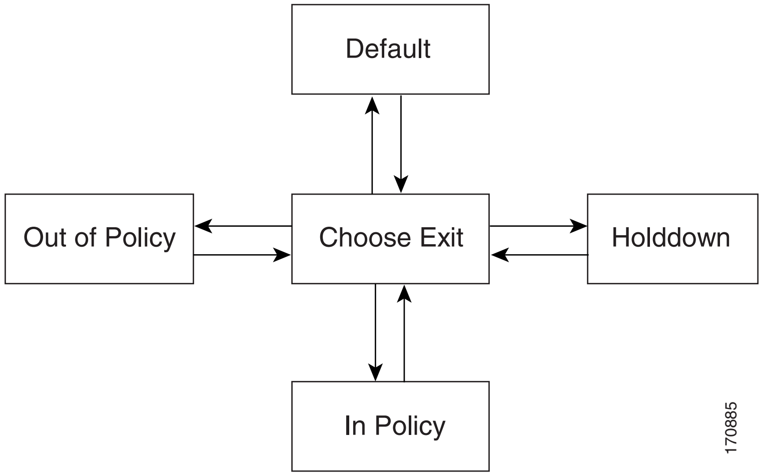
Figure 1 OER Performance Measuring Process



OER measures the performance of both traffic classes and links, but before monitoring a traffic class or link OER checks the state of the traffic class or link. OER uses a policy decision point (PDP) that operates according to the traffic class state transition diagram shown in Figure 2. In some states, OER does not initiate monitoring. The state transition diagram in Figure 2 contains the following states:

- Default—A traffic class is placed in the default state when it is not under OER control. Traffic classes are placed in the default state when they are initially added to the central policy database, the MTC. A traffic class will transition into and out of the default state depending on performance measurements, timers, and policy configuration.
- Choose Exit—This is a temporary state in which the PDP compares the current state of the traffic class against its policy settings and chooses the optimal exit for the traffic class. OER will try to keep a traffic class flowing through its current exit but, as in the default state, performance measurements, timers, and policy configurations can cause the master controller to place a traffic class in this state for the duration of the exit link selection process. The traffic class remains in the choose exit state until it is moved to the new exit.
- Holddown—A traffic class is placed in the holddown state when the master controller requests a border router to forward the traffic class to be monitored using probes. Measurements are collected for the selected traffic class until the holdown timer expires unless the exit used by this traffic class is declared unreachable. If the exit is unreachable, the traffic class transitions back to the choose exit state.

Figure 2 OER Traffic Class State Transition Diagram



- In-Policy—After performance measurements are compared against default or user-defined policy settings and an exit selection is made, the traffic class enters an in-policy state. When a traffic class is in the in-policy state, the traffic class is forwarded through an exit that satisfies the default or user-defined settings. The master controller continues to monitor the traffic class, but no action is taken until the periodic timer expires, or an out-of-policy message is received from a measurement collector, when the traffic class transitions back to the choose exit state.
- Out-of-Policy (OOP)—A traffic class is placed in this state when there are no exits through which to forward the traffic class that conform to default or user-defined policies. While the traffic class is in this state, the backoff timer controls exiting from this state. Each time the traffic class enters this state, the amount of time the traffic class spends in this state increases. The timer is reset for a traffic class when the traffic class enters an in-policy state. If all exit links are out-of-policy, the master controller may select the best available exit.

After determining the state of the traffic class or link, OER may initiate one of the following performance measuring processes:

- OER Traffic Class Performance Measurement, page 4
- OER Link Utilization Measurement, page 10

OER Traffic Class Performance Measurement

OER uses three methods of traffic class performance measurement:

- Passive monitoring—measuring the performance metrics of traffic class entries while the traffic is flowing through the device using NetFlow functionality.
- Active monitoring—creating a stream of synthetic traffic replicating a traffic class as closely as possible and measuring the performance metrics of the synthetic traffic. The results of the performance metrics of the synthetic traffic are applied to the traffic class in the MTC list. Active monitoring uses integrated IP Service Level Agreements (IP SLAs) functionality.
- Both active and passive monitoring—combining both active and passive monitoring in order to generate a more complete picture of traffic flows within the network.

In Cisco IOS Release 12.4(15)T, another variation of the combined active and passive monitoring modes was introduced—fast failover monitoring mode. In fast failover monitoring mode, all exits are continuously probed using active monitoring and passive monitoring. When fast failover monitoring mode is enabled, the probe frequency can be set to a lower frequency than for other monitoring modes, to allow a faster failover capability.

No explicit NetFlow or IP SLAs configuration is required and support for NetFlow and IP SLAs is enabled automatically. You can use both active and passive monitoring methods for a traffic class.

After the master controller is defined and OER functionality is enabled, the master controller uses both passive and active monitoring by default. All traffic classes are passively monitored using integrated NetFlow functionality. Out-of-policy traffic classes are actively monitored using IP SLA functionality. You can configure the master controller to use only passive monitoring, active monitoring, both passive and active monitoring, or fast failover monitoring. The main differences between the different modes can be seen in Table 1.

Table 1 Mode Comparison Table

Comparison Parameter	Active Mode	Passive Mode	Combined Mode	Fast Failover Mode
Release Introduced	12.3(14)T	12.3(14)T	12.3(14)T	12.4(15)T
Active/IP SLA	Yes	No	Yes	Yes
Passive/NetFlow	No	Yes	Yes	Yes
Monitoring of Alternate Paths	On Demand	On Demand	On Demand	Continuous
Best Failover Time	10 seconds	~ 1 minute	~ 1.1 minute	3 seconds
Support for Round Trip Delay	Yes	Yes	Yes	Yes
Support for Loss	Only with Jitter probe	Only for TCP traffic	Only for TCP traffic	Only for TCP traffic and Jitter probe
Support for Reachability	Yes	Only for TCP traffic	Only for TCP traffic	Yes
Support for Jitter	Yes	No	No	Yes
Support for MOS	Yes	No	No	Yes

In Cisco IOS Release 12.2(33)SXH support for using a Cisco Catalyst 6500 series switch as an OER border router was introduced. The master controller that communicates with the Cisco Catalyst 6500 series switch being used as a border router must be a router running Cisco IOS Release 12.4(6)T or a later release because a special monitoring mode was introduced to support the limited capabilities for collecting passive statistics on the Catalyst 6500. The special mode is set globally and cannot be configured using the command-line interface (CLI). For more details, see the "OER Special Monitoring Support for Cisco Catalyst 6500 Series Switches Used as Border Routers" section on page 10.

For more details about each of the monitoring methods, see the following concepts:

- OER Passive Monitoring, page 6
- OER Active Monitoring, page 6
- OER Combined Monitoring, page 9
- OER Fast Failover Monitoring, page 10

OER Passive Monitoring

Cisco IOS OER uses NetFlow, an integrated technology in Cisco IOS software, to collect and aggregate passive monitoring statistics on a per traffic class basis. Passive monitoring is enabled along with active monitoring by default when an OER managed network is created. Passive monitoring can also be enabled explicitly using the **mode monitor passive** command. Netflow is a flow-based monitoring and accounting system, and NetFlow support is enabled by default on the border routers when passive monitoring is enabled.

Passive monitoring uses only existing traffic; additional traffic is not generated. Border routers collect and report passive monitoring statistics to the master controller approximately once per minute. If traffic does not go over an external interface of a border router, no data is reported to the master controller. Threshold comparison is done at the master controller. In Cisco IOS Release 12.4(6)T, passive monitoring is supported only for prefixes. In Cisco IOS Release 12.4(9)T, 12.2(33)SRB, and later releases, passive monitoring supports traffic classes defined by prefix, port, protocol, and DSCP value.

OER uses passive monitoring to measure the following metrics for all the traffic classes:

- Delay—OER measures the average delay of TCP flows for a given prefix. Delay is the measurement of the round-trip response time (RTT) between the transmission of a TCP synchronization message and receipt of the TCP acknowledgement.
- Packet loss—OER measures packet loss by tracking TCP sequence numbers for each TCP flow. OER estimates packet loss by tracking the highest TCP sequence number. If a subsequent packet is received with a lower sequence number, OER increments the packet loss counter. Packet loss is measured in packets per million.
- Reachability—OER measures reachability by tracking TCP synchronization messages that have been sent repeatedly without receiving a TCP acknowledgement.
- Throughput—OER measures throughput by measuring the total number of bytes and packets for each traffic class for a given interval of time.

Note Although all traffic classes are monitored, delay, loss, and reachability information is captured only for TCP traffic flows. Throughput statistics are captured for all non-TCP traffic flows.

Passive monitoring of application traffic was introduced in Cisco IOS Release 12.4(9)T, 12.2(33)SRB, and later releases, with application traffic class configuration support of the profiling of DSCP values as well as protocol and port numbers. DSCP values, port numbers, and protocols in addition to prefixes, are all now sent to the master controller. Passive monitoring statistics are gathered and stored in a prefix history buffer that can hold a minimum of 60 minutes of information depending on whether the traffic flow is continuous. OER uses this information to determine if the prefix is in-policy based on the default or user-defined policies. No alternative path analysis is performed as the traffic for a traffic class is flowing through one transit device in the network. If the traffic class goes OOP and only passive monitoring mode is enabled, the traffic class is moved to another point and the measurement repeated until a good or best exit is found. If the traffic class goes OOP and both passive and active monitoring modes are enabled, active probing is executed on all the exits and a best or good exit is selected. For more details on good and best exit selections, see the "Configuring and Applying OER Policies" module.

OER Active Monitoring

If OER passive monitoring techniques create too much overhead on a network device, or the performance metrics of a traffic class cannot be measured using the OER passive monitoring mode, then OER active monitoring techniques are performed. Active monitoring involves creating a stream of synthetic traffic that replicates a traffic class as closely as possible. The performance metrics of the synthetic traffic are measured and the results are applied to the traffic class entry in the MTC list. In Cisco IOS Release 12.4(6)T, and earlier releases, active monitoring supports traffic classes defined by prefix, port, and protocol. In Cisco IOS Release 12.4(9)T, 12.2(33)SRB, and later releases, active monitoring supports traffic classes defined by prefix, port, protocol, and DSCP value.

OER uses active monitoring to measure the following metrics for all the traffic classes:

- Delay—OER measures the average delay of TCP, UDP, and ICMP flows for a given prefix. Delay is the measurement of the round-trip response time (RTT) between the transmission of a TCP synchronization message and receipt of the TCP acknowledgement.
- Reachability—OER measures reachability by tracking TCP synchronization messages that have been sent repeatedly without receiving a TCP acknowledgement.
- Jitter—Jitter means interpacket delay variance. OER measures jitter by sending multiple packets to a target address and a specified target port number, and measuring the delay interval between packets arriving at the destination.
- MOS—Mean Opinion Score (MOS) is a standards-based method of measuring voice quality. Standards bodies like the ITU have derived two important recommendations: P.800 (MOS) and P.861 (Perceptual Speech Quality Measurement [PSQM]). P.800 is concerned with defining a method to derive a Mean Opinion Score of voice quality. MOS scores range between 1 representing the worst voice quality, and 5 representing the best voice quality. A MOS of 4 is considered “toll-quality” voice.

The creation of synthetic traffic in Cisco network devices is activated through the use of Cisco IOS IP SLA probes. OER is integrated with IP SLAs functionality such that OER will use IP SLA probes to actively monitor a traffic class. When active monitoring is enabled, the master controller commands the border routers to send active probes to set of target IP addresses. The border sends probe packets to no more than five target host addresses per traffic class, and transmits the probe results to the master controller for analysis.

IP SLA Active Probe Types Used by OER

IP SLAs are an embedded feature set in Cisco IOS software and they allow you to analyze IP service levels for IP applications and services, to increase productivity, to lower operational costs, and to reduce occurrences of network congestion or outages. IP SLAs use active traffic monitoring—the generation of traffic in a continuous, reliable, and predictable manner—for measuring network performance. The accuracy of measured data is enhanced by enabling the IP SLAs Responder, available in Cisco routers, on the destination device. For more details about IP SLAs, see the *Cisco IOS IP SLAs Configuration Guide*.

The following types of active probes can be configured:

- ICMP Echo—A ping is sent to the target address. OER uses ICMP Echo probes, by default, when an active probe is automatically generated. Configuring an ICMP echo probe does not require knowledgeable cooperation from the target device. However, repeated probing could trigger an Intrusion Detection System (IDS) alarm in the target network. If an IDS is configured in a target network that is not under your control, we recommend that you notify the administrator of this target network.
- Jitter—A jitter probe is sent to the target address. A target port number must be specified. A remote responder must be enabled on the target device, regardless of the configured port number. Jitter probe support was introduced in Cisco IOS Release 12.4(6)T and 12.2(33)SRB. In Cisco IOS Release 12.4(15)T support for loss policy was introduced for active monitoring if the jitter probe is used.

- TCP Connection—A TCP connection probe is sent to the target address. A target port number must be specified. A remote responder must be enabled if TCP messages are configured to use a port number other than TCP port number 23, which is well-known.
- UDP Echo—A UDP echo probe is sent to the target address. A target port number must be specified. A remote responder must be enabled on the target device, regardless of which port number is configured.

In Cisco IOS Release 12.4(9)T, 12.2(33)SRB, and later releases, OER marks the probe packets with the DSCP value by default if the monitored traffic classes have the DCSP field set to a nonzero value.

Creation of Active Probe for a Traffic Class

To create an active probe for a traffic class, a probe type has to be discovered, and a probe target assigned to the traffic class. To discover a probe type, OER uses one of the following methods:

- Learned probe—Active probes are automatically generated when a traffic class is learned using the NetFlow TopTalker Learn mechanism. Five targets are learned for each traffic class and, by default, the active probe is set as an ICMP echo probe.
- Configured probe—Active probes can also be configured on the master controller by specifying the probe type, target address and port if needed. Configured traffic classes can be configured to use any of the IP SLA active probes.

To assign a probe target for a traffic class, OER uses one of the following methods:

- Longest match—By default, OER assigns a probe target to the traffic class with the longest matching prefix in the MTC list. This is referred to as a default probe assignment.
- Forced assignment—An IP SLA probe can be configured using an OER map and the results of the probe are assigned to specific traffic classes associated with the OER map. This specific assignment of active probe results is called a forced target probe assignment.

The active probe is sourced from the border router and transmitted through an external interface (the external interface may, or may not, be the preferred route for an optimized prefix). When creating an active probe through an external interface for a specified target, the target should be reachable through the external interface. To test the reachability of the specified target, OER performs a route lookup in the BGP and static routing tables for the specified target and external interface.

In active monitoring mode, the probes are activated from all the border routers to find the best performance path for the specific traffic class. The active probes for that traffic class are not activated again unless the traffic class goes OOP.

In Cisco IOS Release 12.4(4)T and earlier releases, the frequency of an active probe used by OER was set to 60 seconds. In Cisco IOS Release 12.4(6)T and 12.2(33)SRB the frequency can be increased for each policy by configuring a lower time-interval between two probes. Increased probe frequency can reduce the response time and, for voice traffic, provide a better approximation of the MOS-low count percentage.

OER Active Probe Source Address

Support for the ability to configure an OER active probe source address was introduced in Cisco IOS Release 12.4(2)T and 12.2(33)SRB. By default, active probes use the source IP address of the OER external interface that transmits the probe. The active probe source address feature is configured on the border router. When this command is configured, the primary IP address of the specified interface is used as the active probe source. The active probe source interface IP address must be unique to ensure that the probe reply is routed back to the specified source interface. If the interface is not configured with an IP address, the active probe will not be generated. If the IP address is changed after the interface has been configured as an active probe source, active probing is stopped, and then restarted with the new IP address. If the IP address is removed after the interface has been configured as an active probe source, active probing is stopped and not restarted until a valid primary IP address is configured.

OER Voice Traffic Optimization Using Active Probes

In Cisco IOS Release 12.4(6)T and 12.2(33)SRB support was introduced for outbound optimization of voice traffic using active probes on the basis of voice metrics such as delay, reachability, jitter, and Mean Opinion Score (MOS).

OER voice traffic optimization provides support for outbound optimization of voice traffic on the basis of the voice performance metrics such as delay, reachability, jitter, and MOS. Delay, reachability, jitter and MOS are important quantitative quality metrics for voice traffic, and these voice metrics are measured using OER active probes. In Cisco IOS Release 12.4(4)T and earlier releases, OER probes could measure delay and reachability, but not jitter and MOS. The IP SLA jitter probe is integrated with OER to measure jitter (source to destination) and the MOS score in addition to measuring delay and reachability. The jitter probe requires a responder on the remote side just like the UDP Echo probe. Integration of the IP SLA jitter probe type in OER enhances the ability of OER to optimize voice traffic. OER policies can be configured to set the threshold and priority values for the voice performance metrics: delay, reachability, jitter, and MOS.

Configuring an OER policy to measure jitter involves configuring only the threshold value and not relative changes (used by other OER features) because for voice traffic, relative jitter changes have no meaning. For example, jitter changes from 5 milliseconds to 25 milliseconds are just as bad in terms of voice quality as jitter changes from 15 milliseconds to 25 milliseconds. If the short-term average (measuring the last 5 minutes) jitter is higher than the jitter threshold, the prefix is considered out-of-policy due to jitter. OER then probes all exits, and the exit with the least jitter is selected as the best exit.

MOS policy works in a different way. There is no meaning to average MOS values, but there is meaning to the number of times that the MOS value is below the MOS threshold. For example, if the MOS threshold is set to 3.85 and if 3 out of 10 MOS measurements are below the 3.85 MOS threshold, the MOS-low-count is 30 percent. When OER runs a policy configured to measure MOS, both the MOS threshold value and the MOS-low-count percentage are considered. A prefix is considered out-of-policy if the short term (during the last 5 minutes) MOS-low-count percentage is greater than the configured value for a given MOS threshold. OER then probes all exits, and the exit with the highest MOS value is selected as the best exit.

OER Combined Monitoring

Cisco IOS OER can also be configured to combine both active and passive monitoring in order to generate a more complete picture of traffic flows within the network. There are some scenarios in which you may want to combine both OER monitoring modes.

One example scenario is when you want to learn traffic classes and then monitor them passively, but you also want to determine the alternate path performance metrics in order to control the traffic classes. The alternate path performance metrics, in the absence of the actual traffic flowing through the alternate path in the network, can be measured using the active probes. OER automates this process by learning traffic classes at five targets and probing through all the alternate paths using active probes.

OER Fast Failover Monitoring

In Cisco IOS Release 12.4(15)T, a new monitoring mode, fast monitoring, was introduced. Fast monitoring sets the active probes to continuously monitor all the exits (probe-all), and passive monitoring is enabled too. Fast failover monitoring can be used with all types of active probes: ICMP echo, Jitter, TCP connection, and UDP echo. When the **mode monitor fast** command is enabled, the probe frequency can be set to a lower frequency than for other monitoring modes, to allow a faster failover ability. Under fast monitoring with a lower probe frequency, route changes can be performed within 3 seconds of an out-of-policy situation. When an exit becomes OOP under fast monitoring, the select best exit is operational and the routes from the OOP exit are moved to the best in-policy exit. Fast monitoring is a very aggressive mode that incurs a lot of overhead with the continuous probing. We recommend that you use fast monitoring only for performance sensitive traffic. For example, a voice call is very sensitive to any performance problems or congested links, but the ability to detect and reroute the call within a few seconds can demonstrate the value of using fast monitoring mode.

OER Special Monitoring Support for Cisco Catalyst 6500 Series Switches Used as Border Routers

In Cisco IOS Release 12.2(33)SXH support for using a Cisco Catalyst 6500 series switch as an OER border router was introduced. Only border router functionality is included in the Cisco IOS Release 12.2(33)SXH images; no master controller configuration is available. The master controller that communicates with the Cisco Catalyst 6500 series switch being used as a border router must be a router running Cisco IOS Release 12.4(6)T or a later release.

In Cisco IOS Release 12.4(6)T the OER master controller software was modified to support the limited capabilities for collecting passive statistics on a Cisco Catalyst 6500 switch used as a border router. If mode monitor active is configured on the master controller, no changes are made. If mode monitor passive or mode monitor both is configured, the master controller sends commands to each border router to determine if the border router can activate passive monitoring. If the master controller has mode monitor passive configured, a Catalyst 6500 border router will be disconnected because it cannot activate passive monitoring. If mode monitor both is configured on the master controller and at least one border router cannot activate passive monitoring then the master controller changes the mode to a special mode. The special mode is set globally and cannot be configured using the command-line interface (CLI). In the special mode only a subset of passive performance metrics—the ingress and egress bandwidth—are evaluated for a traffic class. Active monitoring at regular intervals using a periodic timer supplies the delay and reachability statistics.

When the special monitoring mode is set, the PDP examines probing results for delay and unreachability statistics when measuring the performance of a traffic class. Bandwidth calculations are considered, but loss is not supported.

OER Link Utilization Measurement

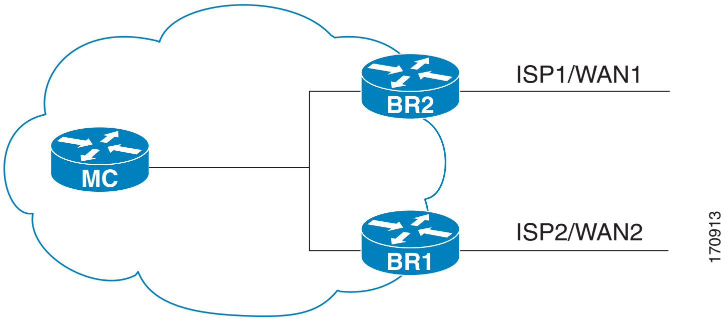
Link Utilization Threshold

After an external interface is configured for a border router, OER automatically monitors the utilization of the external link (an external link is an interface on a border router that typically links to a WAN). Every 20 seconds, by default, the border router reports the link utilization to the master controller. In Cisco IOS Release 12.4(6)T and prior releases, only egress (transmitted) traffic utilization values were reported, but in Cisco IOS Release 12.4(9)T, 12.2(33)SRB, and later releases, the ingress (received) traffic utilization values are also reported to the master controller. If the exit or entrance link utilization is above the default threshold of 75 percent, the exit or entrance link is in an OOP state and OER starts the monitoring process to find an alternative link for the traffic class. The link utilization threshold can be manually configured either as an absolute value in kilobytes per second (kbps) or as a percentage.

Link Utilization Range

OER can also be configured to calculate the range of utilization over all the links. In Cisco IOS Release 12.4(6)T and prior releases, only egress (transmitted) traffic utilization range values were reported, but in Cisco IOS Release 12.4(9)T, 12.2(33)SRB, and later releases, the ingress (received) traffic utilization range values are also reported to the master controller. In Figure 3 there are two border routers with exits links to the Internet through two ISPs. The master controller determines which link on one of the border routers—either BR1 or BR2 in Figure 3—is used by a traffic class.

Figure 3 OER network diagram



OER range functionality attempts to keep the exit or entrance links within a utilization range, relative to each other to ensure that the traffic load is distributed. The range is specified as a percentage and is configured on the master controller to apply to all the exit or entrance links on border routers managed by the master controller. For example, if the range is specified as 25 percent, and the utilization of the exit link at BR1 (in Figure 3) is 70 percent, then if the utilization of the exit link at BR2 (in Figure 3) falls to 40 percent, the percentage range between the two exit links will be more than 25 percent and OER will attempt to move some traffic classes to use the exit link at BR1 to even the traffic load. If BR1 (in Figure 3) is being configured as an entrance link, the link utilization range calculations work in the same way as for an exit link, except that the utilization values are for received traffic, not transmitted traffic.

How to Measure the Traffic Class Performance and Link Utilization Using OER

This section contains the following tasks:

Modifying the OER Link Utilization for Outbound Traffic

Perform this task at the master controller to modify the OER exit (outbound) link utilization threshold. After an external interface has been configured for a border router, OER automatically monitors the utilization of external links on a border router every 20 seconds. The utilization is reported back to the master controller and, if the utilization exceeds 75 percent, OER selects another exit link for traffic classes on that link. An absolute value in kilobytes per second (kbps), or a percentage, can be specified.

To modify the link utilization threshold for inbound traffic, see the "Modifying the OER Link Utilization for Inbound Traffic" section on page 14.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **oer master**
4. **border** *ip-address* [**key-chain** *key-chain-name*]
5. **interface** *type number* **external**
6. **max-xmit-utilization** {**absolute** *kbps* | **percentage** *value*}
7. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`oer master`**
	Example:
	`Router(config)# oer master`
		Enters OER master controller configuration mode to configure a router as a master controller and to configure global operations and policies.

	Command or Action	Purpose
Step 4	**border** *ip-address* [**key-chain** *key-chain-name*] **Example:** `Router(config-oer-mc)# border 10.1.1.2`	Enters OER-managed border router configuration mode to establish communication with a border router. • An IP address is configured to identify the border router. • At least one border router must be specified to create an OER-managed network. A maximum of ten border routers can be controlled by a single master controller. **Note** The **key-chain** keyword and *key-chain-name* argument must be entered when a border router is initially configured. However, this keyword is optional when reconfiguring an existing border router.
Step 5	**interface** *type number* **external** **Example:** `Router(config-oer-mc-br)# interface Ethernet 1/0 external`	Configures a border router interface as an OER-managed external interface and enters OER border exit interface configuration mode. • External interfaces are used to forward traffic and for active monitoring. • A minimum of two external border router interfaces are required in an OER-managed network. At least one external interface must be configured on each border router. A maximum of 20 external interfaces can be controlled by single master controller. **Note** Entering the **interface** command without the **external** or **internal** keyword places the router in global configuration mode and not OER border exit configuration mode. The **no** form of this command should be applied carefully so that active interfaces are not removed from the router configuration. Only the syntax relevant to this task is displayed. For more details, see the *Cisco IOS Optimized Edge Routing Command Reference*.
Step 6	**max-xmit-utilization** {**absolute** *kbps* | **percentage** *value* **Example:** `Router(config-oer-mc-br-if)# max-xmit-utilization absolute 500000`	Configures the maximum utilization on a single OER managed exit link. • Use the **absolute** keyword and *kbps* argument to specify the absolute maximum utilization on an OER managed exit link in kbps. • Use the **percentage** keyword and *value* argument to specify percentage utilization of an exit link.

Step 7	**end**
	Example:
	`Router(config-oer-mc-br-if)# end`
		Exits OER border exit interface configuration mode and returns to privileged EXEC mode.

Modifying the OER Link Utilization for Inbound Traffic

In Cisco IOS Release 12.4(9)T, 12.2(33)SRB, and later releases, the ability to report inbound traffic utilization to the master controller was introduced. Perform this task at the master controller to modify the OER entrance (inbound) link utilization threshold. After an external interface has been configured for a border router, OER automatically monitors the utilization of entrance links on a border router every 20 seconds. The utilization is reported back to the master controller and, if the utilization exceeds 75 percent, OER selects another entrance link for traffic classes on that link. An absolute value in kilobytes per second (kbps), or a percentage, can be specified. This task is configured in the same way as the "Modifying the OER Link Utilization for Outbound Traffic" section on page 12 as an external interface can be used as either an exit link or an entrance link. The difference in the configuration for this task is the command that specifies the utilization threshold for inbound traffic.

Prerequisites

This task requires the master controller and border routers to be running Cisco IOS Release 12.4(9)T, 12.2(33)SRB, or later release.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **oer master**
4. **border** *ip-address* [**key-chain** *key-chain-name*]
5. **interface** *type number* **external**
6. **maximum utilization receive** {**absolute** *kbps* | **percent** *percentage*}
7. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	`enable`
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	`configure terminal`
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	`oer master`
	Example:
	`Router(config)# oer master`
		Enters OER master controller configuration mode to configure a router as a master controller and to configure global operations and policies.

	Command or Action	Purpose

Step 4	**`border`** *`ip-address`* `[`**`key-chain`** *`key-chain-name`*`]`
	Example:
	`Router(config-oer-mc)# border 10.1.1.2`
		Enters OER-managed border router configuration mode to establish communication with a border router.
		• An IP address is configured to identify the border router.
		• At least one border router must be specified to create an OER-managed network. A maximum of ten border routers can be controlled by a single master controller.
		Note The **key-chain** keyword and *key-chain-name* argument must be entered when a border router is initially configured. However, this keyword is optional when reconfiguring an existing border router.

Step 5	**`interface`** *`type number`* **`external`**
	Example:
	`Router(config-oer-mc-br)# interface Ethernet 1/0 external`
		Configures a border router interface as an OER-managed external interface and enters OER border exit interface configuration mode.
		• External interfaces are used to forward traffic and for active monitoring.
		• A minimum of two external border router interfaces are required in an OER-managed network. At least one external interface must be configured on each border router. A maximum of 20 external interfaces can be controlled by single master controller.
		Note Entering the **interface** command without the **external** or **internal** keyword places the router in global configuration mode and not OER border exit configuration mode. The **no** form of this command should be applied carefully so that active interfaces are not removed from the router configuration.
		Only the syntax relevant to this task is displayed. For more details, see the *Cisco IOS Optimized Edge Routing Command Reference*.

	Command or Action	Purpose

Step 6	**maximum utilization receive** {**absolute** *kbps* | **percent** *percentage*}
	Example:
	`Router(config-oer-mc-br-if)# maximum utilization receive percent 90`
		Sets the maximum receive utilization threshold for the configured OER-managed link interface.
		• Use the **absolute** keyword and *kbps* argument to specify the absolute threshold value, in kilobytes per second (kbps), of the throughput for all the entrance links.
		• Use the **percent** keyword and *percentage* argument to specify the maximum utilization threshold as a percentage of bandwidth received by all the entrance links.
		• In this example, the maximum utilization threshold of inbound traffic on this entrance link on the border router must be 90 percent, or less.

Step 7	**end**
	Example:
	`Router(config-oer-mc-br-if)# end`
		Exits OER border exit interface configuration mode and returns to privileged EXEC mode.

Modifying the OER Exit Link Utilization Range

Perform this task at the master controller to modify the maximum exit link utilization range threshold over all the border routers. By default, OER automatically monitors the utilization of external links on a border router every 20 seconds, and the border router reports the utilization to the master controller. If the utilization range between all the exit links exceeds 20 percent, the master controller tries to equalize the traffic load by moving some traffic classes to another exit link. The maximum utilization range is configured as a percentage.

OER uses the maximum utilization range to determine if exit links are in-policy. OER will equalize outbound traffic across all exit links by moving traffic classes from overutilized or out-of-policy exits to in-policy exits.

To modify the link utilization range for entrance links, see the "Modifying the OER Entrance Link Utilization Range" section on page 17.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **oer master**
4. **max-range-utilization percent** *maximum*
5. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`oer master`**
	Example:
	`Router(config)# oer master`
		Enters OER master controller configuration mode to configure a router as a master controller and to configure global operations and policies.

Step 4	**`max-range-utilization percent`** *`maximum`*
	Example:
	`Router(config-oer-mc)# max-range-utilization percent 25`
		Sets the maximum utilization range for all OER-managed exit link.s.
		• Use the **percent** keyword and *maximum* argument to specify the maximum utilization range between all the exit links.
		• In this example, the utilization range between all the exit links on the border routers must be within 25 percent.

Step 5	**`end`**
	Example:
	`Router(config-oer-mc)# end`
		Exits OER master controller configuration mode and returns to privileged EXEC mode.

Modifying the OER Entrance Link Utilization Range

Perform this task at the master controller to modify the maximum entrance link utilization range over all the border routers. By default, OER automatically monitors the utilization of external links on a border router every 20 seconds, and the border router reports the utilization to the master controller. In Cisco IOS Release 12.4(9)T, 12.2(33)SRB, and later releases, the ability to report inbound traffic utilization to the master controller, and to specify a link utilization range for entrance links, was introduced. In this task, if the utilization range between all the entrance links exceeds 20 percent, the master controller tries to equalize the traffic load by moving some traffic classes to another entrance link. The maximum utilization range is configured as a percentage.

OER uses the maximum utilization range to determine if links are in-policy. In this task, OER will equalize inbound traffic across all entrance links by moving traffic classes from overutilized or out-of-policy exits to in-policy exits.

Prerequisites

This task requires the master controller and border routers to be running Cisco IOS Release 12.4(9)T, 12.2(33)SRB, or later release.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **oer master**
4. **max range receive percent** *percentage*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**oer master** **Example:** `Router(config)# oer master`	Enters OER master controller configuration mode to configure a router as a master controller and to configure global operations and policies.
Step 4	**max range receive percent** *percentage* **Example:** `Router(config-oer-mc)# max range receive percent 20`	Specifies the upper limit of the receive utilization range between all the entrance links on the border routers. • The **percent** keyword and *percentage* argument are used to specify the range percentage. • In this example, the receive utilization range between all the entrance links on the border routers must be within 20 percent.

Step 5	**end**
	Example:
	`Router(config-oer-mc)# end`
		Exits OER master controller configuration mode and returns to privileged EXEC mode.

Configuring and Verifying OER Passive Monitoring

OER enables passive monitoring by default when an OER managed network is created, but there are times when passive monitoring is disabled. Use this task to configure passive monitoring and then verify that the passive monitoring is being performed. Perform this task on a border router to display passive measurement information collected by NetFlow for monitored prefixes or application traffic flows. These commands are entered on a border router through which the application traffic is flowing. The **show** commands can be entered in any order.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **oer master**
4. **mode monitor** {**active** | **both** | **passive**}
5. end
6. **show oer border passive cache** {**applications** | **learned** [**application**] | **prefix**}
7. **show oer border passive prefixes**

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

```
Router> enable
```

Step 2 **configure terminal**

Enters global configuration mode.

```
Router# configure terminal
```

Step 3 **oer master**

Enters OER master controller configuration mode to configure a router as a master controller and to configure global operations and policies.

```
Router(config)# oer master
```

Step 4 **mode monitor** {**active** | **both** | **passive**}

Configures route monitoring or route control on an OER master controller. The **monitor** keyword is used to configure active monitoring, passive monitoring, or both active and passive monitoring. Passive monitoring is enabled when either the **both** or **passive** keywords are specified. In this example, passive monitoring is enabled.

Note Only the syntax relevant to this task is displayed. For more details, see the *Cisco IOS Optimized Edge Routing Command Reference*.

```
Router(config-oer-mc)# mode monitor passive
```

Step 5 **end**

Exits OER master controller configuration mode and returns to privileged EXEC mode.

```
Router(config-oer-mc)# end
```

Step 6 **show oer border passive cache** {**applications** | **learned** [**application**] | **prefix**}

This command is used to display real-time passive measurement information collected by NetFlow from the border router for OER monitored prefixes and traffic flows. The **applications** keyword displays information about the monitored application traffic classes, and the **prefix** keyword displays information about monitored prefixes. Using the **learned** and **application** keywords you can display information about learned applications. The following output shows the passive measurement information collected by NetFlow for monitored prefixes and traffic flows for the border router on which the **show oer border passive cache prefix** command was run:

```
Router# show oer border passive cache prefix

OER Passive Prefix Cache, State: enabled, 278544 bytes
  1 active, 4095 inactive, 2 added
  82 ager polls, 0 flow alloc failures
  Active flows timeout in 1 minutes
  Inactive flows timeout in 15 seconds
IP Sub Flow Cache, 17416 bytes
  2 active, 1022 inactive, 4 added, 2 added to flow
  0 alloc failures, 0 force free
  1 chunk, 2 chunks added

Prefix              NextHop         Src If         Dst If
                     Flows   Pkts   B/Pk  Active   sDly   #Dly  PktLos #UnRch
------------------------------------------------------------------------------
10.1.5.0/24         10.1.2.2        Et0/0          Et1/0
                      381    527      40    65.5    300      2       10       1
```

The following output shows the passive measurement information collected by NetFlow for monitored application traffic flows for the border router on which the **show oer border passive cache applications** command was run:

```
Router# show oer border passive cache applications

OER Passive Prefix Cache, State: enabled, 278544 bytes
  6 active, 4090 inactive, 384 added
  6438 ager polls, 0 flow alloc failures
  Active flows timeout in 1 minutes
  Inactive flows timeout in 15 seconds
IP Sub Flow Cache, 25800 bytes
  18 active, 1006 inactive, 1152 added, 384 added to flow
  0 alloc failures, 0 force free
  1 chunk, 1 chunk added

Prefix              NextHop         Src If         Dst If          Flows
 Prot  DSCP SrcPort         DstPort          Appl_ID
                      Pkts   B/Pk  Active   sDly   #Dly  PktLos  #UnRch
------------------------------------------------------------------------
10.1.1.0/24         10.1.1.2        Et8/0          Et0/0              1
17       ef [1, 65535]      [3000, 4000]     2
                        2      28    16.5      0      0       0       0
10.1.3.0/24         10.1.1.2        Et8/0          Et0/0              1
17       ef [1, 65535]      [3000, 4000]     1
                       16      28    19.9      0      0       0       0
```

Step 7 **show oer border passive prefixes**

This command is used to display passive measurement information collected by NetFlow for OER monitored prefixes and traffic flows. The following output shows the prefix that is being passively monitored by NetFlow for the border router on which the **show oer border passive prefixes** command was run:

```
Router# show oer border passive prefixes

OER Passive monitored prefixes:

Prefix         Mask   Match Type
10.1.5.0       /24     exact
```

Configuring OER Active Probing Using the Longest Match Target Assignment

Perform this task at the master controller to configure active probing using the longest match target assignment. Active monitoring is enabled with the **mode monitor active** or **mode monitor both** commands, and the type of active probe is specified using the **active-probe** command. Active probes are configured with a specific host or target address and the active probes are sourced on the border router. The active probe source external interface may, or may not, be the preferred route for an optimized prefix. In this example, both active and passive monitoring are enabled and the target IP address of 10.1.5.1 is to be actively monitored using Internet Control Message Protocol (ICMP) echo (ping) messages. This task does not require an IP SLA responder to be enabled.

OER Active Probing Target Reachability

The active probe is sourced from the border router and transmitted through an external interface (the external interface may or may not be the preferred route for an optimized prefix). When creating an active probe through an external interface for a specified target, the target should be reachable through the external interface. To test the reachability of the specified target, OER performs a route lookup in the BGP and static routing tables for the specified target and external interface.

ICMP Echo Probes

Configuring an ICMP echo probe does not require knowledgeable cooperation from the target device. However, repeated probing could trigger an IDS alarm in the target network. If an IDS is configured in a target network that is not under your administrative control, we recommend that you notify the target network administration entity.

The following defaults are applied when active monitoring is enabled:

- The border router collects up to five host addresses from the traffic class for active probing when a traffic class is learned or aggregated.
- Active probes are sent once per minute.
- ICMP probes are used to actively monitor learned traffic classes.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **oer master**
4. **mode monitor** {**active** | **both** | **passive**}
5. **active-probe** {**echo** *ip-address* | **tcp-conn** *ip-address* **target-port** *number* | **udp-echo** *ip-address* **target-port** *number*}
6. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`oer master`**
	Example:
	`Router(config)# oer master`
		Enters OER master controller configuration mode to configure a router as a master controller and to configure global operations and policies.

Step 4	`mode monitor {active | both | passive}`
	Example:
	`Router(config-oer-mc)# mode monitor both`
		Configures route monitoring on an OER master controller.
		• The **monitor** keyword is used to configure active and/or passive monitoring.
		• The example enables both active and passive monitoring.
		Note Only the syntax relevant to this task is displayed. For more details, see the *Cisco IOS Optimized Edge Routing Command Reference*.

	Command or Action	Purpose

Step 5	**active-probe** {**echo** *ip-address* | **tcp-conn** *ip-address* **target-port** *number* | **udp-echo** *ip-address* **target-port** *number*}
	Example:
	`Router(config-oer-mc)# active-probe echo 10.1.5.1`
		Configures an active probe for a target prefix.
		• Active probing measures delay and jitter of the target prefix more accurately than is possible with only passive monitoring.
		• Active probing requires you to configure a specific host or target address.
		• Active probes are sourced from an OER managed external interfaces. This external interface may or may not be the preferred route for an optimized prefix.
		• A remote responder with the corresponding port number must be configured on the target device when configuring UDP echo probe or when configuring a TCP connection probe that is configured with a port number other than 23. The remote responder is configured with the **ip sla monitor responder** global configuration command.
		Note The **ip sla monitor responder** command was introduced in Cisco IOS Release 12.3(14)T and 12.2(33)SRB. This command replaces the **rtr responder** command.

Step 6	`end`
	Example:
	`Router(config-oer-mc)#` **`end`**
		Exits OER master controller configuration mode and returns to privileged EXEC mode.

Configuring OER Voice Probes with a Forced Target Assignment

Perform this task to enable active monitoring using OER jitter probes. Support for the jitter probe was introduced in Cisco IOS Release 12.4(6)T and 12.2(33)SRB. In this example, the traffic to be monitored is voice traffic, which is identified using an access list. The active voice probes are assigned a forced target for OER instead of the usual longest match assigned target. This task also demonstrates how to modify the OER probe frequency, another feature added in Cisco IOS Release 12.4(6)T and 12.2(33)SRB.

Before configuring the OER jitter probe on the source device, the IP SLAs Responder must be enabled on the target device (the operational target). The IP SLAs Responder is available only on Cisco IOS software-based devices. Start this task at the network device that runs the IP SLAs Responder.

Note The device that runs the IP SLAs Responder does not have to be configured for OER.

Prerequisites

This task requires the master controller and border routers to be running Cisco IOS Release 12.4(6)T, 12.2(33)SRB, or later releases.

Jitter

Jitter means interpacket delay variance. When multiple packets are sent consecutively from source to destination, for example, 10 ms apart, and if the network is behaving ideally, the destination should be receiving them 10 ms apart. But if there are delays in the network (like queuing, arriving through alternate routes, and so on) the arrival delay between packets might be greater than or less than 10 ms. Using this example, a positive jitter value indicates that the packets arrived more than 10 ms apart. If the packets arrive 12 ms apart, then positive jitter is 2 ms; if the packets arrive 8 ms apart, then negative jitter is 2 ms. For delay-sensitive networks like VoIP, positive jitter values are undesirable, and a jitter value of 0 is ideal.

MOS

Mean Opinion Score (MOS) is a quantitative quality metric for voice traffic that can be measured using OER active probes. With all the factors affecting voice quality, many people ask how voice quality can be measured. Standards bodies like the ITU have derived two important recommendations: P.800 (MOS) and P.861 (Perceptual Speech Quality Measurement [PSQM]). P.800 is concerned with defining a method to derive a Mean Opinion Score of voice quality. MOS scores range between 1 representing the worst voice quality, and 5 representing the best voice quality. A MOS of 4 is considered “toll-quality” voice.

Prerequisites

Before configuring this task, an access list must be defined. For an example access list and more details about configuring voice traffic using active probes, see the “OER Voice Traffic Optimization Using Active Probes” solution module.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla monitor responder**
4. **exit**
5. Move to the network device that is the OER master controller.
6. **enable**
7. **configure terminal**
8. oer master
9. **mode monitor** {**active** | **both** | **passive**}
10. exit
11. **oer-map** *map-name sequence-number*
12. **match ip address** {**access-list** *access-list-name* | **prefix-list** *prefix-list-name*}
13. **set active probe** *probe-type ip-address* [**target-port** *number*] [**codec** *codec-name*]
14. **set probe frequency** *seconds*
15. **end**
16. **show oer master active-probes forced**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. - Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip sla monitor responder** **Example:** `Router(config)# ip sla monitor responder`	Enables the IP SLAs Responder.
Step 4	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 5	Move to the network device that is the OER master controller.	—
Step 6	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. - Enter your password if prompted.
Step 7	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 8	**oer master** **Example:** `Router(config)# oer master`	Enters OER master controller configuration mode to configure a router as a master controller and to configure global operations and policies.

Step 9	**mode monitor** {**active** | **both** | **passive**}
	Example:
	`Router(config-oer-mc)# mode monitor active`
		Configures route monitoring on an OER master controller.
		- The **monitor** keyword is used to configure active and/or passive monitoring.
		- The example enables active monitoring.
		Note Only the syntax relevant to this task is displayed. For more details, see the *Cisco IOS Optimized Edge Routing Command Reference*.

Step 10	**exit** **Example:** `Router(config-oer-mc)# exit`	Exits OER master controller configuration mode and returns to global configuration .

	Command or Action	Purpose
Step 11	**oer-map** *map-name sequence-number* **Example:** `Router(config)# oer-map TARGET_MAP 10`	Enters OER map configuration mode to configure an OER map to apply policies to selected IP prefixes. • *Only one match clause can be configured for each OER map sequence.* • Deny sequences are first defined in an IP prefix list and then applied with the **match ip address** (OER) command in Step 12. • The example creates an OER map named TARGET_MAP.

Step 12	**match ip address** {**access-list** *access-list-name* | **prefix-list** *prefix-list-name*}
	Example:
	`Router(config-oer-map)# match ip address access-list VOICE_ACCESS_LIST`
		References an extended IP access list or IP prefix as match criteria in an OER map.
		• Only a single match clause can be configured for each OER map sequence.
		• The example configures the IP access list named VOICE_ACCESS_LIST as match criteria in an OER map.

Step 13	**set active-probe** *probe-type ip-address* [**target-port** *number*] [**codec** *codec-name*] **Example:** `Router(config-oer-map)# set active-probe jitter 10.20.22.1 target-port 2000 codec g729a`	Creates a set clause entry to assign a target prefix for an active probe. • Use the *probe-type* argument to specify one four probe types: echo, jitter, tcp-conn, or udp-echo. • The *ip-address* argument to specify the target IP address of a prefix to be monitored using the specified type of probe. • The **target-port** keyword and *number* argument are used to specify the destination port number for the active probe. • The **codec** keyword and *codec-name* argument are used only with the jitter probe type. Specifies the codec value used for Mean Opinion Score (MOS) calculation. The codec values must be one of the following: g711alaw, g711ulaw, or g729a. • The example creates a set clause entry to specify the target IP address of a prefix and a specific port number to actively monitor using jitter.
Step 14	**set probe frequency** *seconds* **Example:** `Router(config-oer-map)# set probe frequency 10`	Creates a set clause entry to set the frequency of the OER active probe. • The *seconds* argument is used to set the time, in seconds, between the active probe monitoring of the specified IP prefixes. • The example creates a set clause to set the active probe frequency to 10 seconds.

	Command or Action	Purpose
Step 15	**set jitter threshold** *maximum* **Example:** `Router(config-oer-map)# set jitter threshold 20`	Creates a set clause entry to configure the jitter threshold value. • The **threshold** keyword is used to configure the maximum jitter value, in milliseconds. • The example creates a set clause that sets the jitter threshold value to 20 for traffic that is matched in the same OER map sequence.

Step 16	**set mos** {**threshold** *minimum* **percent** *percent*}
	Example:
	`Router(config-oer-map)# set mos threshold 4.0 percent 30`
		Creates a set clause entry to configure the MOS threshold and percentage values used to decide whether an alternate exit is be selected.
		• The **threshold** keyword is used to configure the minimum MOS value.
		• The **percent** keyword is used to configure the percentage of MOS values that are below the MOS threshold.
		• OER calculates the percentage of MOS values below the MOS threshold that are recorded in a five-minute period. If the percentage value exceeds the configured percent value or the default value, the master controller searches for alternate exit links.
		• The example creates a set clause that sets the threshold MOS value to 4.0 and the percent value to 30 percent for traffic that is matched in the same OER map sequence.

Step 17	**set delay** {**relative** *percentage* | **threshold** *maximum*}
	Example:
	`Router(config-oer-map)# set delay threshold 100`
		Creates a set clause entry to configure the delay threshold.
		• The delay threshold can be configured as a relative percentage or as an absolute value for match criteria.
		• The **relative** keyword is used to configure a relative delay percentage. The relative delay percentage is based on a comparison of short-term and long-term measurements.
		• The **threshold** keyword is used to configure the absolute maximum delay period in milliseconds.
		• The example creates a set clause that sets the absolute maximum delay threshold to 100 milliseconds for traffic that is matched in the same OER map sequence.

Step 18	**end**
	Example:
	`Router(config-oer-map)# end`
		Exits OER map configuration mode and enters privileged EXEC mode.

	Command or Action	Purpose
Step 19	**show oer master active-probes** [**appl** | **forced**] **Example:** `Router# show oer master active-probes forced`	Displays connection and status information about active probes on an OER master controller. • The output from this command displays the active probe type and destination, the border router that is the source of the active probe, the target prefixes that are used for active probing, and whether the probe was learned or configured. • The **appl** keyword is used to filter the output to display information about applications optimized by the master controller. • The **forced** keyword is used to show any forced targets that are assigned. • The example displays connection and status information about the active probes generated for voice traffic configured with a forced target assignment.

Examples

This example shows output from the **show oer master active-probes forced** command. The output is filtered to display only connection and status information about the active probes generated for voice traffic configured with a forced target assignment.

```
Router# show oer master active-probes forced

OER Master Controller active-probes
Border   = Border Router running this Probe
Policy   = Forced target is configure under this policy
Type     = Probe Type
Target   = Target Address
TPort    = Target Port
N - Not applicable

The following Forced Probes are running:

Border          State    Policy             Type     Target          TPort
10.20.20.2     ACTIVE    40                 jitter   10.20.22.1      3050
10.20.21.3     ACTIVE    40                 jitter   10.20.22.4      3050
```

Configuring OER Voice Probes for Fast Failover

In Cisco IOS Release 12.4(15)T the ability to configure a fast monitoring mode was introduced. In fast failover monitoring mode, all exits are continuously probed using active monitoring and passive monitoring. The probe frequency can be set to a lower frequency in fast failover monitoring mode than for other monitoring modes, to allow a faster failover capability. Fast failover monitoring can be used with all types of active probes: ICMP echo, jitter, TCP connection, and UDP echo.

Perform this task to enable fast monitoring using OER jitter probes. Fast failover monitoring is designed for traffic classes that are very sensitive to performance issues or congested links, and voice traffic is very sensitive to any dropped links. In this example, the fast failover monitoring mode is enabled and the voice traffic to be monitored is identified using an IP prefix list. To reduce some of the overhead that fast failover monitoring produces, the active voice probes are assigned a forced target for OER. The OER probe frequency is set to 2 seconds. In the examples section after the task table, the **show oer master prefix** command is used to show the policy configuration for the prefix specified in the task steps and some logging output is displayed to show that fast failover is configured.

Note Fast monitoring is a very aggressive mode that incurs a lot of overhead with the continuous probing. We recommend that you use fast monitoring only for performance sensitive traffic.

Before configuring the OER jitter probe on the source device, the IP SLAs Responder must be enabled on the target device (the operational target). The IP SLAs Responder is available only on Cisco IOS software-based devices. Start this task at the network device that runs the IP SLAs Responder.

Note The device that runs the IP SLAs Responder does not have to be configured for OER.

Prerequisites

This task requires the master controller and border routers to be running Cisco IOS Release 12.4(15)T, or later releases.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla monitor responder**
4. **exit**
5. Move to the network device that is the OER master controller.
6. **enable**
7. **configure terminal**
8. **ip prefix-list**-*name* [**seq** *seq-value*] {**deny** *network/length* | **permit** *network/length*} [**le** *le-value*]
9. Repeat Step 8 for more prefix list entries, as required.
10. **oer-map** *map-name sequence-number*
11. **match traffic-class prefix-list** *prefix-list-name*
12. **set mode monitor** {**active** | **both** | **fast** | **passive**}
13. **set jitter threshold** *maximum*
14. **set mos** {**threshold** *minimum* **percent** *percent*}
15. **set delay** {**relative** *percentage* | **threshold** *maximum*}
16. **set active probe** *probe-type ip-address* [**target-port** *number*] [**codec** *codec-name*]
17. **set probe frequency** *seconds*
18. **end**
19. **show oer master prefix** [*prefix* [**detail** | **policy** | **traceroute** [*exit-id* | *border-address* | **current**]]]

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`ip sla monitor responder`**
	Example:
	`Router(config)# ip sla monitor responder`
		Enables the IP SLAs Responder.

Step 4	**`exit`**
	Example:
	`Router(config)# exit`
		Exits global configuration mode and returns to privileged EXEC mode.

Step 5	Move to the network device that is the OER master controller.	—

Step 6	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 7	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 8	**ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network/length* | **permit** *network/length*} [**le** *le-value*] **Example:** `Router(config)# ip prefix-list VOICE_FAIL_LIST permit 10.1.0.0/24`	Creates an IP prefix list. • The IP prefix list specified here is used in an OER map to specify the destination IP addresses for a traffic class. • The example creates an IP prefix list named VOICE_FAIL_LIST for OER to profile the prefix, 10.1.0.0/24.
Step 9	Repeat Step 4 for more prefix list entries, as required.	—
Step 10	**oer-map** *map-name sequence-number* **Example:** `Router(config)# oer-map FAST_FAIL_MAP 10`	Enters OER map configuration mode to configure an OER map to apply policies to selected IP prefixes. • *Only one match clause can be configured for each OER map sequence.* • The example creates an OER map named FAST_FAIL_MAP.

	Command or Action	Purpose

Step 11	**`match traffic-class prefix-list`** *`prefix-list-name`*
	Example:
	`Router(config-oer-map)# match traffic-class prefix-list VOICE_FAIL_LIST`
		References an IP prefix list as traffic class match criteria in an OER map.
		• Only a single match clause can be configured for each OER map sequence.
		• The example configures the IP prefix list named VOICE_FAIL_LIST as match criteria in an OER map.

Step 12	**`set mode monitor`** `{`**`active`** `|` **`both`** `|` **`fast`** `|` **`passive`**`}`
	Example:
	`Router(config-oer-map)# set mode monitor fast`
		Creates a set clause entry to configure route monitoring on an OER master controller.
		• The **monitor** keyword is used to configure active and/or passive monitoring.
		• The **fast** keyword is used to configure fast failover monitoring mode where continuous active monitoring is enabled as well as passive monitoring.
		• The example enables fast failover monitoring.
		Note Only the syntax relevant to this task is displayed. For more details, see the *Cisco IOS Optimized Edge Routing Command Reference*.

Step 13	**`set jitter threshold`** *`maximum`*
	Example:
	`Router(config-oer-map)# set jitter threshold 12`
		Creates a set clause entry to configure the jitter threshold value.
		• The **threshold** keyword is used to configure the maximum jitter value, in milliseconds.
		• The example creates a set clause that sets the jitter threshold value to 12 for traffic that is matched in the same OER map sequence.

Step 14	**`set mos`** `{`**`threshold`** *`minimum`* **`percent`** *`percent`*`}`
	Example:
	`Router(config-oer-map)# set mos threshold 3.6 percent 30`
		Creates a set clause entry to configure the MOS threshold and percentage values used to decide whether an alternate exit is be selected.
		• The **threshold** keyword is used to configure the minimum MOS value.
		• The **percent** keyword is used to configure the percentage of MOS values that are below the MOS threshold.
		• OER calculates the percentage of MOS values below the MOS threshold that are recorded in a five-minute period. If the percentage value exceeds the configured percent value or the default value, the master controller searches for alternate exit links.
		• The example creates a set clause that sets the threshold MOS value to 3.6 and the percent value to 30 percent for traffic that is matched in the same OER map sequence.

	Command or Action	Purpose

Step 15	**set delay** {**relative** *percentage* | **threshold** *maximum*}
	Example:
	`Router(config-oer-map)# set delay relative 50`
		Creates a set clause entry to configure the delay threshold.
		• The delay threshold can be configured as a relative percentage or as an absolute value for match criteria.
		• The **relative** keyword is used to configure a relative delay percentage. The relative delay percentage is based on a comparison of short-term and long-term measurements.
		• The **threshold** keyword is used to configure the absolute maximum delay period in milliseconds.
		• The example creates a set clause that sets the relative delay percentage to 50 percent for traffic that is matched in the same OER map sequence.

Step 16	**set active-probe** *probe-type ip-address* [**target-port** *number*] [**codec** *codec-name*] **Example:** `Router(config-oer-map)# set active-probe jitter 10.120.120.1 target-port 20 codec g729a`	Creates a set clause entry to assign a target prefix for an active probe. • Use the *probe-type* argument to specify one four probe types: echo, jitter, tcp-conn, or udp-echo. • The *ip-address* argument to specify the target IP address of a prefix to be monitored using the specified type of probe. • The **target-port** keyword and *number* argument are used to specify the destination port number for the active probe. • The **codec** keyword and *codec-name* argument are used only with the jitter probe type. Specifies the codec value used for Mean Opinion Score (MOS) calculation. The codec values must be one of the following: g711alaw, g711ulaw, or g729a. • The example creates a set clause entry to specify the target IP address of a prefix and a specific port number to actively monitor using jitter.
Step 17	**set probe frequency** *seconds* **Example:** `Router(config-oer-map)# set probe frequency 2`	Creates a set clause entry to set the frequency of the OER active probe. • The *seconds* argument is used to set the time, in seconds, between the active probe monitoring of the specified IP prefixes. • The example creates a set clause to set the active probe frequency to 2 seconds. **Note** A probe frequency of less than 4 seconds is possible here because the fast failover monitoring mode has been enabled in Step 12.

Step 18	**end**
	Example:
	`Router(config-oer-map)# end`
		Exits OER map configuration mode and enters privileged EXEC mode.

	Command or Action	Purpose
Step 19	**show oer master prefix** [*prefix* [**detail** | **policy** | **traceroute** [*exit-id* | *border-address* | **current**]]] **Example:** `Router# show oer master prefix 10.1.1.0/24 policy`	(Optional) Displays the status of monitored prefixes. - The *prefix* argument is entered as an IP address and bit length mask. - The **policy** keyword is used to display policy information for the specified prefix. - The example displays policy information for the prefix, 10.1.1.0/24. **Note** Only the syntax relevant to this task is displayed. For more details, see the *Cisco IOS Optimized Edge Routing Command Reference*.

Examples

This example shows output from the **show oer master prefix** command when a prefix is specified with the policy keyword to display the policy configured for the prefix 10.1.1.0/24. Note that the mode monitor is set to fast, which automatically sets the select-exit to best, and allows the probe frequency to be set at 2.

```
Router# show oer master prefix 10.1.1.0/24 policy

* Overrides Default Policy Setting
oer-map MAP 10
  sequence no. 8444249301975040, provider id 1, provider priority 30
    host priority 0, policy priority 10, Session id 0
  match ip prefix-lists: VOICE_FAIL_LIST
  backoff 90 90 90
  delay relative 50
  holddown 90
  periodic 0
 *probe frequency 2
  mode route control
 *mode monitor fast
 *mode select-exit best
  loss relative 10
 *jitter threshold 12
  mos threshold 3.60 percent 30
  unreachable relative 50
  next-hop not set
  forwarding interface not set
  resolve jitter priority 1 variance 10
  resolve utilization priority 12 variance 20

  Forced Assigned Target List:
   active-probe jitter 10.120.120.1 target-port 20 codec g729a
```

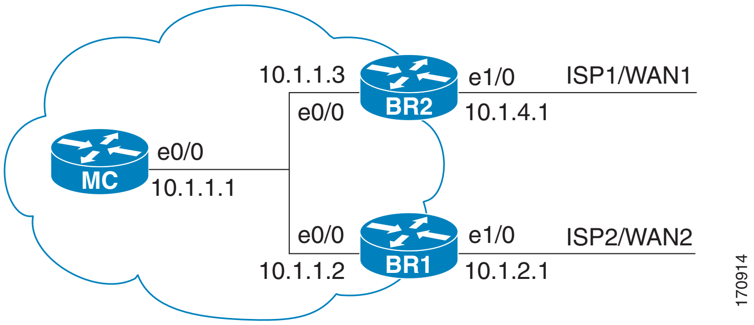
After the master controller is configured for fast failover as shown in this task, and a traffic class goes out of policy, the logging output below shows that the traffic class represented by prefix 10.1.1.0 is routed by OER through a new border router exit at interface 10.3.3.4 within 3 seconds. From the logging output, it appears that the traffic class moved to an out-of-policy state due to the jitter threshold being exceeded.

```
May  2 10:55:27.355: %OER_MC-5-NOTICE: Active ABS Jitter OOP Prefix 10.1.1.0/24,
jitter 15, BR 10.4.4.2, i/f Et2/0
May  2 10:55:27.367: %OER_MC-5-NOTICE: Route changed Prefix 10.1.1.0/24, BR 10.3.3.4,
i/f Et5/0, Reason Jitter, OOP Reason Jitter
```

Configuring Exit Link Load Balancing Using OER

Perform this task at the master controller to configure load balancing for traffic classes over the border router exit links. In this example, both active and passive monitoring is enabled, and range and exit utilization policies are given priority when OER chooses the best exit selection for traffic classes. Best route selection for performance policies is disabled. The external Ethernet interfaces on border router 1 and border router 2—BR1 and BR2 in Figure 4—are both configured with a maximum utilization threshold of 70 percent. After an external interface is configured for the border routers, OER automatically monitors the utilization of external links on a border router every 20 seconds. The utilization is reported back to the master controller and, if the utilization exceeds 70 percent, OER selects another exit link for traffic classes on that link.

Figure 4 Network diagram for OER Exit Link Load Balancing



Traffic can also be load balanced over entrance links, for more details see the “Using OER to Control Traffic Classes and Verify the Network Performance” module.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **oer master**
4. **mode monitor** {**active** | **both** | **passive**}
5. **resolve range priority** *value*
6. **resolve utilization priority** *value* **variance** *percentage*
7. **no resolve delay**
8. **no resolve loss**
9. **border** *ip-address* [**key-chain** *key-chain-name*]
10. **interface** *type number* **external**
11. **max-xmit-utilization** {**absolute** *kbps* | **percentage** *value*}
12. **exit**
13. **exit**
14. Repeat Step 8 through Step 12 for each border router.
15. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`oer master`**
	Example:
	`Router(config)# oer master`
		Enters OER master controller configuration mode to configure a router as a master controller and to configure global operations and policies.

Step 4	**mode monitor** {**active** | **both** | **passive**}
	Example:
	`Router(config-oer-mc)# mode monitor both`
		Configures route monitoring on an OER master controller.
		• The **monitor** keyword is used to configure active and/or passive monitoring.
		• The example enables both active and passive monitoring.
		Note Only the syntax relevant to this task is displayed. For more details, see the *Cisco IOS Optimized Edge Routing Command Reference*.

Step 5	**resolve range priority** *value* **Example:** `Router(config-oer-mc)# resolve range priority 1`	Sets policy priority or resolves policy conflicts. • This command is used to set priority when multiple policies are configured for the same prefix. When this command is configured, the policy with the highest priority will be selected to determine the policy decision. • The **priority** keyword is used to specify the priority value. Setting the number 1 assigns the highest priority to a policy. Setting the number 10 assigns the lowest priority. • Each policy must be assigned a different priority number. • In this example, the priority for range policies is set to 1. **Note** Only the syntax relevant to this task is displayed. For more details, see the *Cisco IOS Optimized Edge Routing Command Reference*.

	Command or Action	Purpose

Step 6	**`resolve utilization priority`** *`value`* **`variance`** *`percentage`*
	Example:
	`Router(config-oer-mc)# resolve utilization priority 2 variance 25`
		Sets policy priority or resolves policy conflicts.
		• This command is used to set priority when multiple policies are configured for the same prefix. When this command is configured, the policy with the highest priority will be selected to determine the policy decision.
		• The **priority** keyword is used to specify the priority value. Setting the number 1 assigns the highest priority to a policy. Setting the number 10 assigns the lowest priority.
		• Each policy must be assigned a different priority number.
		• The **variance** keyword is used to set an allowable variance for a user-defined policy. This keyword configures the allowable percentage that an exit link or prefix can vary from the user-defined policy value and still be considered equivalent.
		• In this example, the priority for range policies is set to 2 with a 25 percent variance.
		Note Only the syntax relevant to this task is displayed. For more details, see the *Cisco IOS Optimized Edge Routing Command Reference*.

Step 7	**`no resolve delay`**
	Example:
	`Router(config-oer-mc)# no resolve delay`
		Sets policy priority or resolves policy conflicts.
		• This command is used to set priority when multiple policies are configured for the same prefix. When this command is configured, the policy with the highest priority will be selected to determine the policy decision.
		• The example disables the priority for delay performance policies.
		Note Only the syntax relevant to this task is displayed. For more details, see the *Cisco IOS Optimized Edge Routing Command Reference*.

Step 8	**`no resolve loss`**
	Example:
	`Router(config-oer-mc)# no resolve loss`
		Sets policy priority or resolves policy conflicts.
		• This command is used to set priority when multiple policies are configured for the same prefix. When this command is configured, the policy with the highest priority will be selected to determine the policy decision.
		• The example disables the priority for loss performance policies.
		Note Only the syntax relevant to this task is displayed. For more details, see the *Cisco IOS Optimized Edge Routing Command Reference*.

	Command or Action	Purpose

Step 9	**`border`** *`ip-address`* `[`**`key-chain`** *`key-chain-name`*`]`
	
	Example:
	`Router(config-oer-mc)# border 10.1.1.2 key-chain border1_OER`
		Enters OER-managed border router configuration mode to establish communication with a border router.
		• An IP address is configured to identify the border router.
		• At least one border router must be specified to create an OER-managed network. A maximum of ten border routers can be controlled by a single master controller.
		• The value for the *key-chain-name* argument must match a valid the key-chain name configured on the border router.
		Note The **key-chain** keyword and *key-chain-name* argument must be entered when a border router is initially configured. However, this keyword is optional when reconfiguring an existing border router.

Step 10	**`interface`** *`type number`* **`external`**
	
	Example:
	`Router(config-oer-mc-br)# interface Ethernet 1/0 external`
		Configures a border router interface as an OER-managed external interface.
		• External interfaces are used to forward traffic and for active monitoring.
		• A minimum of two external border router interfaces are required in an OER-managed network. At least one external interface must be configured on each border router. A maximum of 20 external interfaces can be controlled by single master controller.
		Tip Configuring an interface as an OER-managed external interface on a router enters OER border exit interface configuration mode. In this mode, you can configure maximum link utilization or cost-based optimization for the interface.
		Note Entering the **interface** command without the **external** or **internal** keyword places the router in global configuration mode and not OER border exit configuration mode. The **no** form of this command should be applied carefully so that active interfaces are not removed from the router configuration.

Step 11	**`max-xmit-utilization`** `{`**`absolute`** *`kbps`* `|` **`percentage`** *`value`*`}`
	
	Example:
	`Router(config-oer-mc-br-if)# max-xmit-utilization absolute 500000`
		Configures the maximum utilization on a single OER managed exit link.
		• Use the **absolute** keyword and *kbps* argument to specify the absolute maximum utilization on an OER managed exit link in kbps.
		• Use the **percentage** keyword and *value* argument to specify percentage utilization of an exit link.

	Command or Action	Purpose
Step 12	**exit** **Example:** `Router(config-oer-mc-br-if)# exit`	Exits OER-managed border exit interface configuration mode and returns to OER-managed border router configuration mode.
Step 13	Repeat Step 9 through Step 12 with appropriate changes to establish communication with each border router.	—
Step 14	**keepalive** *timer* **Example:** `Router(config-oer-mc)# keepalive 10`	(Optional) Configures the length of time that an OER master controller will maintain connectivity with an OER border router after no keepalive packets have been received. • The example sets the keepalive timer to 10 seconds. The default keepalive timer is 60 seconds.

Step 15	**end**
	Example:
	`Router(config-oer-mc)# end`
		Exits OER master controller configuration mode and returns to privileged EXEC mode.

Step 16	**show running-config** **Example:** `Router# show running-config`	(Optional) Displays the running configuration to verify the configuration entered in this task.

Configuring the Source Address of an Active Probe

Perform this task on a border router to specify the source interface for active probing. Support for configuring a source interface for active probing was introduced in Cisco IOS Release 12.4(2)T and 12.2(33)SRB. The active probe source interface is configured on the border router with the **active-probe address source** in OER border router configuration mode. The active probe source interface IP address must be unique to ensure that the probe reply is routed back to the specified source interface.

The following is default behavior:

- The source IP address is used from the default OER external interface that transmits the active probe when this command is not enabled or if the **no** form is entered.
- If the interface is not configured with an IP address, the active probe will not be generated.
- If the IP address is changed after the interface has been configured as an active probe source, active probing is stopped, and then restarted with the new IP address.
- If the IP address is removed after the interface has been configured as an active probe source, active probing is stopped and not restarted until a valid primary IP address is configured.

Prerequisites

This task requires the master controller and border routers to be running Cisco IOS Release 12.4(2)T, 12.2(33)SRB, or later releases.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **oer border**
4. **active-probe address source interface** *type number*
5. **end**
6. **show oer border active-probes**

DETAILED STEPS

	Command or Action	Purpose

Step 1	`enable`
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	`configure terminal`
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	`oer border`
	Example:
	`Router(config)# oer border`
		Enters OER border router configuration mode to configure a router as a border router.

Step 4	**`active-probe address source interface`** *`type number`*
	Example:
	`Router(config-oer-br)# active-probe address source interface FastEthernet 0/0`
		Configures an interface on a border router as the active-probe source.
		• The example configures interface FastEthernet 0/0 as the source interface.

Step 5	`end`
	Example:
	`Router(config-oer-br)# end`
		Exits OER border router configuration mode and enters privileged EXEC mode.

Step 6	`show oer border active-probes`
	Example:
	`Router# show oer border active-probes`
		Displays connection status and information about active probes on an OER border router.
		• Use this command to verify the configured source IP address.

Examples

This example shows output from the **show oer border active-probes** command. The output is filtered to display only connection and status information about the active probes generated for voice traffic configured with a forced target assignment.

```
Router# show oer border active-probes
```

```
        OER Border active-probes
Type      = Probe Type
Target    = Target IP Address
TPort     = Target Port
Source    = Send From Source IP Address
Interface = Exit interface
Att       = Number of Attempts
Comps   = Number of completions
N - Not applicable

Type     Target          TPort Source          Interface           Att    Comps
udp-echo 10.4.5.1           80 10.0.0.1        FE2/0                 1        0
tcp-conn 10.4.7.1           33 10.0.0.1        FE0/0                 1        0
echo     10.4.9.1            N 10.0.0.1        FE1/0                 2        2
```

Configuration Examples for Measuring the Traffic Class Performance and Link Utilization Using OER

The examples in this section show how to configure OER to measure traffic class performance and link utilization.

- Modifying the OER Link Utilization for Outbound Traffic: Example, page 40
- Modifying the OER Link Utilization for Inbound Traffic: Example, page 40
- Modifying the OER Exit Link Utilization Range: Example, page 41
- Modifying the OER Entrance Link Utilization Range: Example, page 41
- Active Probing: Examples, page 41
- Configuring OER Active Probing Using the Longest Match Target Assignment: Examples, page 42
- Configuring Active Probing with a Forced Target Assignment: Examples, page 43
- Configuring OER Voice Probes for Fast Failover: Example, page 44
- Configuring the Source Address of an Active Probe: Example, page 47

Modifying the OER Link Utilization for Outbound Traffic: Example

The following example shows how to modify the OER exit link utilization threshold. In this example, the exit utilization is set to 80 percent. If the utilization for this exit link exceeds 80 percent, OER selects another exit link for traffic classes that were using this exit link.

```
Router(config)# oer master
Router(config-oer-mc)# border 10.1.4.1
Router(config-oer-mc-br)# interface Ethernet 1/0 external
Router(config-oer-mc-br-if)# max-xmit-utilization percentage 80
Router(config-oer-mc-br-if)# end
```

Modifying the OER Link Utilization for Inbound Traffic: Example

The following example shows how to modify the OER entrance link utilization threshold. In this example, the entrance utilization is set to 65 percent. If the utilization for this exit link exceeds 65 percent, OER selects another entrance link for traffic classes that were using this entrance link.

```
Router(config)# oer master
Router(config-oer-mc)# border 10.1.2.1
Router(config-oer-mc-br)# interface Ethernet 1/0 external
Router(config-oer-mc-br-if)# maximum receive utilization percentage 65
Router(config-oer-mc-br-if)# end
```

Modifying the OER Exit Link Utilization Range: Example

The following example shows how to modify the OER exit utilization range. In this example, the exit utilization range for all exit links is set to 10 percent. OER uses the maximum utilization range to determine if exit links are in-policy. OER will equalize outbound traffic across all exit links by moving prefixes from overutilized or out-of-policy exits to in-policy exits.

```
Router(config)# oer master
Router(config-oer-mc)# max-range-utilization percentage 10
Router(config-oer-mc)# end
```

Modifying the OER Entrance Link Utilization Range: Example

The following example shows how to modify the OER entrance utilization range. In this example, the entrance utilization range for all entrance links is set to 15 percent. OER uses the maximum utilization range to determine if entrance links are in-policy. OER will equalize inbound traffic across all entrance links by moving prefixes from overutilized or out-of-policy exits to in-policy exits.

```
Router(config)# oer master
Router(config-oer-mc)# max range receive percent 15
Router(config-oer-mc)# end
```

Active Probing: Examples

ICMP Echo Example

The following example, starting in global configuration mode, configures an active probe using an ICMP echo (ping) message. The 10.5.5.55 address is the target. No explicit configuration is required on the target device.

```
Router(config)# oer master
Router(config-oer-mc)# active-probe echo 10.5.5.55
```

TCP Connection Example

The following example, starting in global configuration mode, configures an active probe using a TCP connection message. The 10.5.55.56 address is the target. The target port number must be specified when configuring this type of probe.

```
Router(config)# oer master
Router(config-oer-mc)# active-probe tcp-conn 10.5.5.56 target-port 23
```

Note A remote responder is required for TCP connection probes when a port other than 23 is configured.

UDP Echo Example

The following example, starting in global configuration mode, configures an active probe using UDP echo messages. The 10.5.5.57 address is the target. The target port number must be specified when configuring this type of probe, and a remote responder must also be enabled on the target device.

```
Router(config)# oer master
Router(config-oer-mc)# active-probe udp-echo 10.5.5.57 target-port 1001
```

UDP Remote Responder Example

The following example, starting in global configuration mode, configures a remote responder on a border router to send IP SLAs control packets in response to UDP active probes. The port number must match the number that is configured for the active probe.

```
Border-Router(config)# ip sla monitor responder type udpEcho port 1001
```

TCP Remote Responder Example

The following example, starting in global configuration mode, configures a remote responder on a border router to send IP SLAs control packets in response to TCP active probes. The remote responder must be configured for TCP active probes that do not use the TCP well-known port number 23.

```
Border-Router(config)# ip sla monitor responder type tcpConnect port 49152
```

Configuring OER Active Probing Using the Longest Match Target Assignment: Examples

The example configurations in this section demonstrate active probing using the longest match target assignment using the following probe types:

- ICMP Probe for Longest Match Target Assignment, page 42
- TCP Probe for Longest Match Target Assignment, page 42
- UDP Probe for Longest Match Target Assignment, page 43

ICMP Probe for Longest Match Target Assignment

The following example shows how to configure active probing using the ICMP probe with the longest match target assignment:

```
Router(config)# oer master
Router(config-oer-mc)# mode monitor active
Router(config-oer-mc)# active-probe echo 10.5.5.55
```

TCP Probe for Longest Match Target Assignment

The following example shows how to configure active probing using the TCP probe with the longest match target assignment. The IP SLAs Responder must first be enabled on the target device, and this device does not have to be configured for OER. A border router can be used as the target device. The second configuration is performed at the master controller.

Target Device

```
Router> enable
Router# configure terminal
Router(config)# ip sla monitor responder type tcpConnect port 49152
Router(config)# exit
```

Master Controller

```
Router(config)# oer master
Router(config-oer-mc)# mode monitor active
Router(config-oer-mc)# active-probe tcp-conn 10.4.4.44 target-port 49152
```

UDP Probe for Longest Match Target Assignment

The following example shows how to configure active probing using the UDP probe with the longest match target assignment. The IP SLAs Responder must first be enabled on the target device, and this device does not have to be configured for OER. A border router can be used as the target device. The second configuration is performed at the master controller.

Target Device

```
Router> enable
Router# configure terminal
Router(config)# ip sla monitor responder type udpEcho port 1001
Router(config)# exit
```

Master Controller

```
Router(config)# oer master
Router(config-oer-mc)# mode monitor active
Router(config-oer-mc)# active-probe udp-echo 10.3.3.33 target-port 1001
```

Configuring Active Probing with a Forced Target Assignment: Examples

The example configurations in this section demonstrate active probing using a forced target assignment using the following probe types:

- UDP Probe for Forced Target Assignment, page 43
- Jitter Probe for Forced Target Assignment, page 44

UDP Probe for Forced Target Assignment

The following example shows how to configure active probing with a forced target assignment and a configured probe frequency of 20 seconds. This example requires an IP SLAs Responder to be enabled on the target device.

Target Device

```
Router> enable
Router# configure terminal
Router(config)# ip sla monitor responder type udpEcho port 1001
Router(config)# exit
```

Master Controller

```
Router(config)# oer master
Router(config-oer-mc)# mode monitor active
Router(config-oer-mc)# exit
Router(config)# oer-map FORCED_MAP 10
Router(config-oer-map)# match ip address access-list FORCED_LIST
Router(config-oer-map)# set active-probe udp-echo 10.5.5.57 target-port 1001
Router(config-oer-map)# set probe frequency 20
Router(config-oer-map)# end
```

Jitter Probe for Forced Target Assignment

The following example shows how to configure active probing for Voice traffic with a forced target assignment using the jitter probe and a configured probe frequency of 15 seconds. The voice traffic is identified using an access list and thresholds are set for jitter, mos, and delay. In this task, the **codec** keyword and *codec-name* argument used in the jitter probe configuration specify the codec value used for mos calculation. This example requires an IP SLAs Responder to be enabled on the target device.

Target Device

```
Router> enable
Router# configure terminal
Router(config)# ip sla monitor responder
Router(config)# exit
```

Master Controller

```
Router(config)# oer master
Router(config-oer-mc)# mode monitor active
Router(config-oer-mc)# exit
Router(config)# oer-map FORCED_VOICE_MAP 10
Router(config-oer-map)# match ip address access-list FORCED_VOICE_LIST
Router(config-oer-map)# set active-probe jitter 172.17.5.57 target-port 2000 codec g729a
Router(config-oer-map)# set probe frequency 15
Router(config-oer-map)# set jitter threshold 20
Router(config-oer-map)# set mos threshold 4.0 percent 30
Router(config-oer-map)# set delay threshold 100
Router(config-oer-map)# end
```

Configuring OER Voice Probes for Fast Failover: Example

The following example, starting in global configuration mode, shows how quickly a new exit can be selected when fast failover is configured.

Note Fast monitoring is a very aggressive mode that incurs a lot of overhead with the continuous probing. We recommend that you use fast monitoring only for performance sensitive traffic.

The first output shows the configuration at the master controller of three border routers. Route control mode is enabled.

```
Router# show run | sec oer master

oer master
 policy-rules MAP
 port 7777
 logging
 !
 border 10.3.3.3 key-chain key1
  interface Ethernet9/0 external
  interface Ethernet8/0 internal
 !
 border 10.3.3.4 key-chain key2
  interface Ethernet5/0 external
  interface Ethernet8/0 internal
 !
 border 10.4.4.2 key-chain key3
  interface Ethernet2/0 external
  interface Ethernet8/0 internal
 backoff 90 90
 mode route control
 resolve jitter priority 1 variance 10
 no resolve delay
!
```

To verify the basic configuration and show the status of the border routers, the **show oer master** command is run:

```
Router# show oer master

OER state: ENABLED and ACTIVE
  Conn Status: SUCCESS, PORT: 7777
  Version: 2.1
  Number of Border routers: 3
  Number of Exits: 3
  Number of monitored prefixes: 1 (max 5000)
  Max prefixes: total 5000 learn 2500
  Prefix count: total 1, learn 0, cfg 1

Border           Status   UP/DOWN             AuthFail  Version
10.4.4.2        ACTIVE   UP       17:00:32          0  2.1
10.3.3.4        ACTIVE   UP       17:00:35          0  2.1
10.3.3.3        ACTIVE   UP       17:00:38          0  2.1

Global Settings:
  max-range-utilization percent 20 recv 20
  mode route metric bgp local-pref 5000
  mode route metric static tag 5000
  trace probe delay 1000
  logging

Default Policy Settings:
  backoff 90 90 90
  delay relative 50
  holddown 90
  periodic 0
  probe frequency 56
  mode route control
  mode monitor both
  mode select-exit good
  loss relative 10
  jitter threshold 20
  mos threshold 3.60 percent 30
  unreachable relative 50
  resolve jitter priority 1 variance 10
  resolve utilization priority 12 variance 20

Learn Settings:
  current state : DISABLED
  time remaining in current state : 0 seconds
  no throughput
  no delay
  no inside bgp
  no protocol
  monitor-period 5
  periodic-interval 120
  aggregation-type prefix-length 24
  prefixes 100
  expire after time 720
```

Fast failover is now configured for active voice probes and the probe frequency is set to 2 seconds using an OER map. The fast failover monitoring mode is enabled and the voice traffic to be monitored is identified using an IP prefix list to specify the 10.1.1.0/24 prefix. To reduce some of the overhead that fast failover monitoring produces, the active voice probes are assigned a forced target for OER.

```
Router# show run | sec oer-map

oer-map MAP 10
 match traffic-class prefix-list VOICE_FAIL_LIST
 set mode select-exit best
 set mode monitor fast
 set jitter threshold 12
 set active-probe jitter 120.120.120.1 target-port 20 codec g729a
 set probe frequency 2
```

The following output from the **show oer master prefix** command when a prefix is specified with the policy keyword shows the policy configured for the prefix 10.1.1.0/24. Note that the mode monitor is set to fast, which automatically sets the select-exit to best, and allows the probe frequency to be set at 2.

```
Router# show oer master prefix 10.1.1.0/24 policy

* Overrides Default Policy Setting
oer-map MAP 10
  sequence no. 8444249301975040, provider id 1, provider priority 30
    host priority 0, policy priority 10, Session id 0
  match ip prefix-lists: VOICE_FAIL_LIST
  backoff 90 90 90
  delay relative 50
  holddown 90
  periodic 0
 *probe frequency 2
  mode route control
 *mode monitor fast
 *mode select-exit best
  loss relative 10
 *jitter threshold 12
  mos threshold 3.60 percent 30
  unreachable relative 50
  next-hop not set
  forwarding interface not set
  resolve jitter priority 1 variance 10
  resolve utilization priority 12 variance 20

  Forced Assigned Target List:
   active-probe jitter 10.120.120.1 target-port 20 codec g729a
```

After the master controller is configured for fast failover as shown in this task, and a traffic class goes out of policy, the logging output below shows that the traffic class represented by prefix 10.1.1.0/24 is routed by OER through a new border router exit at interface 10.3.3.4 within 3 seconds. From the logging output, it appears that the traffic class moved to an out-of-policy state due to the jitter threshold being exceeded.

```
May  2 10:55:27.355: %OER_MC-5-NOTICE: Active ABS Jitter OOP Prefix 10.1.1.0/24,
jitter 15, BR 10.4.4.2, i/f Et2/0
May  2 10:55:27.367: %OER_MC-5-NOTICE: Route changed Prefix 10.1.1.0/24, BR 10.3.3.4,
i/f Et5/0, Reason Jitter, OOP Reason Jitter
```

Configuring the Source Address of an Active Probe: Example

The following example, starting in global configuration mode, configures FastEthernet 0/0 as the active-probe source interface.

```
Router(config)# oer border
Router(config-oer-br)# active-probe address source interface FastEthernet 0/0
```

Where to Go Next

This module described the OER measure phase and it has assumed that you started with the Cisco IOS Optimized Edge Routing Overview module, followed by the Setting Up OER Network Components module. The measure phase is the second phase in the OER performance loop. To learn more about the other OER phases, read through the other modules in the following list:

- Using OER to Profile the Traffic Classes
- Measuring the Traffic Class Performance and Link Utilization Using OER
- Configuring and Applying OER Policies
- Using OER to Control Traffic Classes and Verify the Route Control Changes

After you understand the various OER phases, you may want to review the OER Solutions modules that are listed under "Related Documents" section on page 47.

Additional References

The following sections provide references related to measuring the traffic class performance and link utilization using OER.

Related Documents

Related Topic	Document Title
Cisco OER technology overview	"Cisco IOS Optimized Edge Routing Overview" module
Concepts and configuration tasks required to set up OER network components.	"Setting Up OER Network Components" module
OER solution module: voice traffic optimization using OER active probes.	"OER Voice Traffic Optimization Using Active Probes" module
OER solution module: configuring VPN IPsec/GRE tunnel interfaces as OER-managed exit links.	"Configuring VPN IPsec/GRE Tunnel Interfaces As OER-Managed Exit Links" module
Cisco OER commands: complete command syntax, command mode, command history, defaults, usage guidelines and examples	*Cisco IOS Optimized Edge Routing Command Reference*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Measuring the Traffic Class Performance and Link Utilization Using OER

Table 2 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.3(8)T, 12.2(33)SRB, or a later release appear in the table.

For information on a feature in this technology that is not documented here, see the "Cisco IOS Optimized Edge Routing Feature Roadmap."

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 2 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 2 Feature Information for Measuring the Traffic Class Performance and Link Utilization Using OER

Feature Name	Releases	Feature Configuration Information
Optimized Edge Routing	12.3(8)T 12.2(33)SRB	OER was introduced.
OER Active Probe Source Address	12.4(2)T 12.2(33)SRB	The OER Active Probe Source Address feature allows you to configure a specific exit interface on the border router as the source for active probes. The following sections provide information about this feature: • OER Active Probe Source Address, page 8 • IP SLA Active Probe Types Used by OER, page 7 • Configuring the Source Address of an Active Probe, page 38 • Configuring the Source Address of an Active Probe: Example, page 47 The **active-probe address source** command was introduced by this feature.
OER Voice Traffic Optimization	12.4(6)T 12.2(33)SRB	The OER Voice Traffic Optimization feature introduced support for outbound optimization of voice traffic based on the voice metrics, jitter and Mean Opinion Score (MOS). Jitter and MOS are important quantitative quality metrics for voice traffic and these voice metrics are measured using OER active probes. The following sections provide information about this feature: • OER Voice Traffic Optimization Using Active Probes, page 9 • Configuring OER Voice Probes with a Forced Target Assignment, page 23 The following commands were introduced or modified by this feature: **active-probe**, **jitter**, **mos**, **resolve**, **set jitter**, **set mos**, **set probe**, **set resolve**, **show oer master active-probes**, **show oer policy**, **show oer master prefix**.

Table 2 Feature Information for Measuring the Traffic Class Performance and Link Utilization Using OER

Feature Name	Releases	Feature Configuration Information
OER BGP Inbound Optimization	12.4(9)T 12.2(33)SRB	OER BGP inbound optimization supports best entrance selection for traffic that originates from prefixes outside an autonomous system destined for prefixes inside the autonomous system. External BGP (eBGP) advertisements from an autonomous system to an Internet service provider (ISP) can influence the entrance path for traffic entering the network. OER uses eBGP advertisements to manipulate the best entrance selection. The following sections provide information about this feature: • OER Link Utilization Measurement, page 10 • Modifying the OER Link Utilization for Inbound Traffic, page 14 • Modifying the OER Entrance Link Utilization Range, page 17 • Modifying the OER Link Utilization for Inbound Traffic: Example, page 40 • Modifying the OER Entrance Link Utilization Range: Example, page 41 The following commands were introduced or modified by this feature: **clear oer master prefix**, **downgrade bgp**, **inside bgp**, **match ip address (OER)**, **match oer learn**, **max range receive**, **maximum utilization receive**, **show oer master prefix**.
OER DSCP Monitoring	12.4(9)T 12.2(33)SRB	OER DSCP Monitoring introduced automatic learning of traffic classes based on protocol, port numbers, and DSCP value. Traffic classes can be defined by a combination of keys comprising of protocol, port numbers, and DSCP values, with the ability to filter out traffic that is not required, and the ability to aggregate the traffic in which you are interested. Information such as protocol, port number, and DSCP information is now sent to the master controller database in addition to the prefix information. The new functionality allows OER to both actively and passively monitor application traffic. The following sections provide information about this feature: • OER Passive Monitoring, page 6 • OER Active Monitoring, page 6 The following commands were introduced or modified by this feature: **show oer border passive applications**, **show oer border passive cache**, **show oer border passive learn**, **show oer master appl**, **traffic-class aggregation**, **traffic-class filter**, and **traffic-class keys**.

Table 2 Feature Information for Measuring the Traffic Class Performance and Link Utilization Using OER

Feature Name	Releases	Feature Configuration Information
Support for Fast Failover Monitoring[1]	12.4(15)T	Fast Failover Monitoring introduced the ability to configure a fast monitoring mode. In fast failover monitoring mode, all exits are continuously probed using active monitoring and passive monitoring. The probe frequency can be set to a lower frequency in fast failover monitoring mode than for other monitoring modes, to allow a faster failover capability. Fast failover monitoring can be used with all types of active probes: ICMP echo, jitter, TCP connection, and UDP echo. The following section provides information about this feature: • OER Fast Failover Monitoring, page 10 • Configuring OER Voice Probes for Fast Failover, page 28 • Configuring OER Voice Probes for Fast Failover: Example, page 44 The following commands were modified by this feature: **mode (OER)**, **set mode**.
OER Border Router Only Functionality	12.2(33)SXH	In Cisco IOS Release 12.2(33)SXH support for using a Cisco Catalyst 6500 series switch as an OER border router was introduced. Only border router functionality is included in the Cisco IOS Release 12.2(33)SXH images; no master controller configuration is available. The master controller that communicates with the Cisco Catalyst 6500 series switch being used as a border router must be a router running Cisco IOS Release 12.4(6)T or a later release. The OER master controller software has been modified to handle the limited functionality supported by the Cisco Catalyst 6500 border routers. Using the Route Processor (RP), the Catalyst 6500 border routers can capture throughput statistics only for a traffic class compared to the delay, loss, unreachability, and throughput statistics collected by non-Catalyst 6500 border routers. A master controller automatically detects the limited capabilities of the Catalyst 6500 border routers and downgrades other border routers to capture only the throughput statistics for traffic classes. By ignoring other types of statistics, the master controller is presented with a uniform view of the border router functionality. The following sections provide information about this feature: • OER Traffic Class Performance Measurement, page 4 • OER Special Monitoring Support for Cisco Catalyst 6500 Series Switches Used as Border Routers, page 10 The following command was introduced or modified by this feature: **show oer border passive cache**.

[1] This is a minor enhancement. Minor enhancements are not typically listed in Feature Navigator.

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

Configuring and Applying OER Policies

First Published: January 29, 2007
Last Updated: July 19, 2007

This module describes the Cisco IOS Optimized Edge Routing (OER) apply policy phase. In the apply policy phase, OER uses policies to map the measured performance metrics of traffic class entries in the Monitored Traffic Class (MTC) list, or exit links, against well-known or configured thresholds to determine if the traffic class entry performance or the link utilization is meeting specified levels of service, or if some action is required.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Configuring and Applying OER Policies" section on page 73.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Configuring and Applying OER Policies

Before implementing OER policies, you need to understand an overview of how OER works and how to set up OER network components. See the "Cisco IOS Optimized Edge Routing Overview" and "Setting Up OER Network Components" modules for more details. If you are following the OER performance loop, the OER profile and measure phases precede this phase. See the "Where to Go Next" section on page 71 for more details.

Information About Configuring and Applying OER Policies

To configure and apply OER policies, you should understand the following concepts:

- OER Apply Policy Phase Overview, page 2
- OER Policy Decision Point, page 3
- OER Traffic Class Performance Policies, page 5
- OER Link Policies, page 6
- Performance Routing Link Grouping, page 9
- OER Network Security Policies, page 10
- OER Policy Operational Options and Parameters, page 10
- OER Policy Application, page 12
- Priority Resolution for Multiple OER Policies, page 13

OER Apply Policy Phase Overview

The OER apply policy phase is the third step in the OER performance loop following after the profile phase that identifies the traffic classes, and the measure phase where each traffic class entry in the MTC list is monitored to determine performance metric measurements. The apply policy phase compares the measured performance metrics against well-known or configured thresholds to determine if the traffic is meeting specified levels of service, or if some action is required. If the performance metric does not conform to the threshold, a decision is made by OER to move the traffic class or exit into another state. For more details about the state transition decision, see the "OER Policy Decision Point" section on page 3.

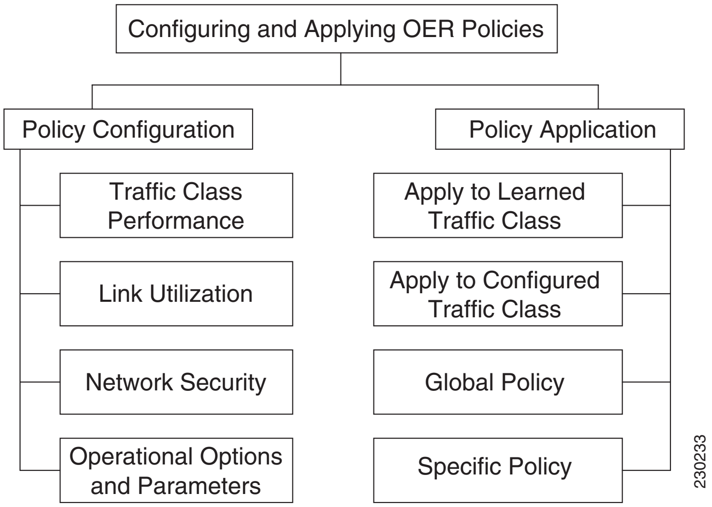
An OER policy is a rule that defines an objective and contains the following attributes:

- A scope.
- An action.
- A triggering event or condition.

For example, a policy can be configured to maintain a delay of less than or equal to 100 milliseconds for packets sent to a specific traffic class entry. The scope is the network traffic sent to the specific traffic class entry, the action is a routing table change, and the triggering event is a measured delay of greater than 100 milliseconds for this traffic. The action may be not be executed until OER is configured to control the traffic in the OER control phase. By default, OER runs in an observe mode during the profile, measure, and apply policy phases.

In the OER apply policy phase you can configure and apply policies. Different types of OER policies can be configured—see Figure 1—and specific OER parameters and options can be included within a policy. In this document, a parameter is a configurable element that can be fine-tuned, and an option is a configurable element that is either enabled or disabled. After an OER policy is configured, the policy can be applied to learned traffic classes or configured traffic classes. OER policies can be applied globally—to all the traffic classes—or to just a specific set of traffic classes.

Figure 1 OER Apply Policy Phase Structure


Configuring and Applying OER Policies
Policy Configuration
Policy Application
Traffic Class Performance
Link Utilization
Network Security
Operational Options and Parameters
Apply to Learned Traffic Class
Apply to Configured Traffic Class
Global Policy
Specific Policy
230233

In Figure 1 you can see that there are three types of OER policies plus some operational options and parameters that can be configured. Use the following links to review more information about each policy type, parameter, or option:

- OER Traffic Class Performance Policies, page 5
- OER Link Policies, page 6
- OER Network Security Policies, page 10
- OER Policy Operational Options and Parameters, page 10

After an OER policy is configured, you can see from Figure 1 that a policy can be applied to learned traffic classes or configured traffic classes on a global basis for all traffic classes or for a specific set of traffic classes. For more details about applying OER policies, see the "OER Policy Application" section on page 12.

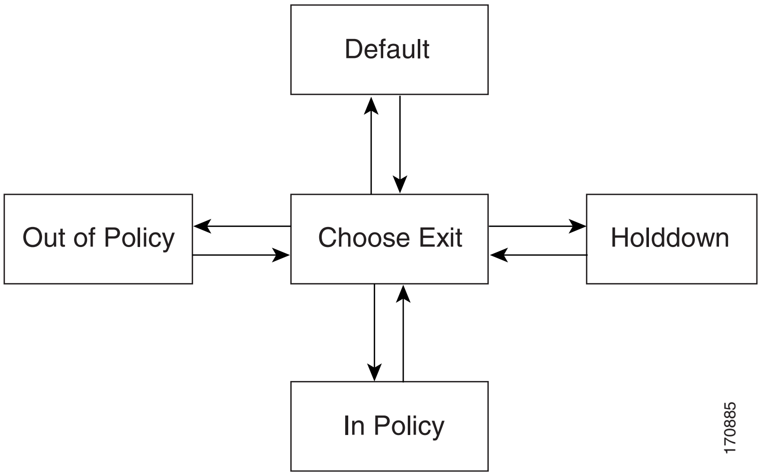
When configuring multiple policy parameters for traffic classes, it is possible to have multiple overlapping policies. To resolve the potential conflict of which policy to run, OER uses its resolve function: a flexible mechanism that allows you to set the priority for most of the policy types. For more details about how OER resolves multiple policy conflicts, see the "Priority Resolution for Multiple OER Policies" section on page 13.

OER Policy Decision Point

When running an OER policy that compares the traffic class performance metrics with default or configured thresholds, a traffic class may change state. OER uses a policy decision point (PDP) that operates according to the traffic class state transition diagram shown in Figure 2. The state transition diagram in Figure 2 contains the following states:

- Default—A traffic class is placed in the default state when it is not under OER control. Traffic classes are placed in the default state when they are initially added to the central policy database, the MTC. A traffic class will transition into and out of the default state depending on performance measurements, timers, and policy configuration.
- Choose Exit—This is a temporary state in which the PDP compares the current state of the traffic class against its policy settings and chooses the optimal exit for the traffic class. OER will try to keep a traffic class flowing through its current exit but, as in the default state, performance measurements, timers, and policy configurations can cause the master controller to place a traffic class in this state for the duration of the exit link selection process. The traffic class remains in the choose exit state until it is moved to the new exit.
- Holddown—A traffic class is placed in the holddown state when the master controller requests a border router to forward the traffic class to be monitored using probes. Measurements are collected for the selected traffic class until the holddown timer expires unless the exit used by this traffic class is declared unreachable. If the exit is unreachable, the traffic class transitions back to the choose exit state.

Figure 2 OER Traffic Class State Transition Diagram



- In-Policy—After performance measurements are compared against default or user-defined policy settings and an exit selection is made, the traffic class enters an in-policy state. When a traffic class is in the in-policy state, the traffic class is forwarded through an exit that satisfies the default or user-defined settings. The master controller continues to monitor the traffic class, but no action is taken until the periodic timer expires, or an out-of-policy message is received from a measurement collector, when the traffic class transitions back to the choose exit state.
- Out-of-Policy (OOP)—A traffic class is placed in this state when there are no exits through which to forward the traffic class that conform to default or user-defined policies. While the traffic class is in this state, the backoff timer controls exiting from this state. Each time the traffic class enters this state, the amount of time the traffic class spends in this state increases. The timer is reset for a traffic class when the traffic class enters an in-policy state. If all exit links are out-of-policy, the master controller may select the best available exit.

OER Traffic Class Performance Policies

OER traffic class performance policies are a set of rules that govern performance characteristics for traffic classes that can be network addresses (prefixes) or application criteria such as protocol, port number, or DSCP value. Network addresses can refer to individual endpoints within a network (e.g. 10.1.1.1/32) or to entire subnets (e.g. 10.0.0.0/8). The major performance characteristics that can be managed within an OER policy are:

- Reachability, page 5
- Delay, page 5
- Packet Loss, page 6
- Jitter, page 6
- Mean Opinion Score (MOS), page 6

With the exception of reachability, none of these performance characteristics can be managed within the constructs of conventional routing protocol metrics. Cisco OER extends the concept of reachability (beyond ensuring that a particular route exists in the routing table) by automatically verifying that the destination can be reached through the indicated path. Using Cisco OER provides the network administrator with a new and powerful toolset for managing the flow of traffic.

Reachability

Reachability is specified as the relative percentage or the absolute maximum number of unreachable hosts, based on flows per million (fpm), that OER will permit from a traffic class entry. If the absolute number or relative percentage of unreachable hosts is greater than the user-defined or the default value, OER determines that the traffic class entry is out-of-policy and searches for an alternate exit link.

To configure parameters for reachability, use the **unreachable** command. This command has two keywords, **relative** and **threshold**. The **relative** keyword is used to configure the relative percentage of unreachable hosts. The relative unreachable host percentage is based on a comparison of short-term and long-term measurements. The short-term measurement reflects the percentage of hosts that are unreachable within a 5-minute period. The long-term measurement reflects the percentage of unreachable hosts within a 60-minute period. The following formula is used to calculate this value:

Relative percentage of unreachable hosts = ((short-term percentage - long-term percentage) / long-term percentage) * 100

The master controller measures the difference between these two values as a percentage. If the percentage exceeds the user-defined or default value, the traffic class entry is determined to be out-of-policy. For example, if 10 hosts are unreachable during the long-term measurement and 12 hosts are unreachable during short-term measurement, the relative percentage of unreachable hosts is 20 percent.

The **threshold** keyword is used to configure the absolute maximum number of unreachable hosts. The maximum value is based on the actual number of hosts that are unreachable based on fpm.

Delay

Delay (also referred as latency) is defined as the delay between when the packet was sent from the source device and when it arrived at a destination device. Delay can be measured as one-way delay or round-trip delay. The largest contributor to latency is caused by network transmission delay.

In Cisco IOS Release 12.4(6)T, 12.2(33)SRB, and later releases, support was introduced for defining delay performance characteristics with respect to voice traffic. Round-trip delay affects the dynamics of conversation and is used in Mean Opinion Score (MOS) calculations. One-way delay is used for diagnosing network problems. A caller may notice a delay of 200 milliseconds and try to speak just as the other person is replying because of packet delay. The telephone industry standard specified in ITU-T G.114 recommends the maximum desired one-way delay be no more than 150 milliseconds. Beyond a one-way delay of 150 milliseconds, voice quality is affected. With a round-trip delay of 300 milliseconds or more, users may experience annoying talk-over effects.

Packet Loss

Packet loss can occur due an interface failing, a packet being routed to the wrong destination, or congestion in the network.

Packet loss for voice traffic leads to the degradation of service in which a caller hears the voice sound with breaks. Although average packet loss is low, voice quality may be affected by a short series of lost packets.

Jitter

Support for jitter was introduced in Cisco IOS Release 12.4(6)T, 12.2(33)SRB, and later releases. Jitter means interpacket delay variance. When multiple packets are sent consecutively from source to destination, for example, 10 ms apart, and if the network is behaving ideally, the destination should be receiving them 10 ms apart. But if there are delays in the network (like queuing, arriving through alternate routes, and so on) the arrival delay between packets might be greater than or less than 10 ms. Using this example, a positive jitter value indicates that the packets arrived more than 10 ms apart. If the packets arrive 12 ms apart, then positive jitter is 2 ms; if the packets arrive 8 ms apart, then negative jitter is 2 ms. For delay-sensitive networks like VoIP, both positive and negative jitter values are undesirable; a jitter value of 0 is ideal.

Mean Opinion Score (MOS)

Support for MOS was introduced in Cisco IOS Release 12.4(6)T, 12.2(33)SRB, and later releases. With all the factors affecting voice quality, many people ask how voice quality can be measured. Standards bodies like the ITU have derived two important recommendations: P.800 (MOS) and P.861 (Perceptual Speech Quality Measurement [PSQM]). P.800 is concerned with defining a method to derive a Mean Opinion Score of voice quality. MOS scores range between 1 representing the worst voice quality, and 5 representing the best voice quality. A MOS of 4 is considered “toll-quality” voice.

In Cisco IOS Release 12.4(4)T and prior releases, only reachability, delay, and loss performance characteristics could be used. In Cisco IOS Release 12.4(6)T, 12.2(33)SRB, and later releases, jitter and MOS performance characteristic can be configured in an OER policy as well as delay and packet loss to determine the quality of a phone call over an IP network.

OER Link Policies

OER link policies are a set of rules that are applied against OER-managed external link (an external link is an interface on a border router on the network edge). Link policies define the desired performance characteristics of the links. Instead of defining the performance of an individual traffic class entry that uses the link (as in traffic class performance policies), link policies are concerned with the performance of the link as a whole. In Cisco IOS Release 12.4(6)T and prior releases, link policies are applied only to exit (egress) links, but in Cisco IOS Release 12.4(9)T, 12.2(33)SRB, and later releases, support for selected entrance (ingress) link policies was introduced. The following performance characteristics are managed by link policies:

- Traffic Load (Utilization)
- Range
- Cost

Traffic Load

A traffic load (also referred to as utilization) policy consists of an upper threshold on the amount of traffic that a specific link can carry. Cisco IOS OER supports per traffic class load distribution. Every 20 seconds, by default, the border router reports the link utilization to the master controller, after an external interface is configured for a border router. In Cisco IOS Release 12.4(6)T and prior releases, only exit link traffic load thresholds can be configured as an OER policy, but in Cisco IOS Release 12.4(9)T, 12.2(33)SRB, and later releases, entrance link traffic load thresholds can be configured. If the exit or entrance link utilization is above the configured threshold, or the default threshold of 75 percent, the exit or entrance link is in an OOP state and OER starts the monitoring process to find an alternative link for the traffic class. The link utilization threshold can be manually configured either as an absolute value in kilobytes per second (kbps) or as a percentage. A load utilization policy for an individual interface is configured on the master controller under the border router configuration.

Tip When configuring load distribution, we recommend that you set the interface load calculation on external interfaces to 30-second intervals with the **load-interval** interface configuration command. The default calculation interval is 300 seconds. The load calculation is configured under interface configuration mode on the border router. This configuration is not required, but it is recommended to allow Cisco IOS OER to respond as quickly as possible to load distribution issues.

Range

A range policy is defined to maintain all links within a certain utilization range, relative to each other in order to ensure that the traffic load is distributed. For example, if a network has multiple exit links, and there is no financial reason to choose one link over another, the optimal choice is to provide an even load distribution across all links. The load-sharing provided by traditional routing protocols is not always evenly distributed, because the load-sharing is flow-based rather than performance- or policy-based. Cisco OER range functionality allows you to configure OER to maintain the traffic utilization on a set of links within a certain percentage range of each other. If the difference between the links becomes too great, OER will attempt to bring the link back to an in-policy state by distributing traffic classes among the available links. The master controller sets the maximum range utilization to 20 percent for all OER-managed links by default, but the utilization range can be configured using a maximum percentage value. In Cisco IOS Release 12.4(6)T and prior releases, only an exit link utilization range can be configured as an OER policy, but in Cisco IOS Release 12.4(9)T, 12.2(33)SRB, and later releases, an entrance link utilization range can be configured.

Cost

OER support for cost-based optimization was introduced in Cisco IOS Release 12.3(14)T and 12.2(33)SRB. Cost-based optimization allow you to configure policies based on the monetary cost (ISP service level agreements [SLAs]) of each exit link in your network. To implement OER cost-based optimization the OER master controller is configured to send traffic over exit links that provide the most cost-effective bandwidth utilization, while still maintaining the desired performance characteristics. Cost-based optimization supports two billing models: fixed-rate billing or tier-based billing.

Fixed-rate billing is used when the ISP bills one flat rate for network access regardless of bandwidth usage. If fixed-rate billing only is configured on the exit links, all exits are considered equal with regard to cost-optimization and other policy parameters (such as delay, loss, and utilization) that are used to determine if the prefix or exit link is in-policy.

If multiple exit links are configured with tiered and fixed policies, then exit links with fixed policies have the highest priority with regard to cost optimization. If the fixed exit links are at maximum utilization, then the tiered exit links will be used.

Tier-based billing is used when the ISP bills at a tiered rate based on the percentage of exit link utilization. Each cost tier is configured separately with an associated monetary cost and a percentage of bandwidth utilization that activates the tier is defined. An allowance is made for bursting in the algorithm used to determine the tier-based billing. Bursting is defined as short periods of high bandwidth usage that would be expensive under fixed-rate billing.

The specific details of tier-based billing models vary by ISP. However, most ISPs use some variation of the following algorithm to calculate what an enterprise should pay in a tiered billing plan:

- Gather periodic measurements of egress and ingress traffic carried on the enterprise connection to the ISP network and aggregate the measurements to generate a rollup value for a rollup period.
- Generate one or more rollup values per billing period.
- Rank the rollup values for the billing period into a stack from the largest value to the smallest.
- Discard the top 5 percent of the rollup values from the stack to accommodate bursting.
- Apply the highest remaining rollup value in the stack to a tiered structure to determine a tier associated with the rollup value.
- Charge the customer based on a set cost associated with the determined tier.

Note A billing policy must be configured and applied to prefixes in order for the master controller to perform cost-based optimization.

At the end of each billing cycle the top n percent of samples, or rollup values, are discarded. The remaining highest value is the sustained utilization. Based on the number of samples discarded, the billing cycle is divided into three periods: the initial period, the middle period, and the last period.

The initial period is when the number of samples measured is less than the number of discards +1. For example, if the discard percentage is 7 percent, billing month is 30 days long, and sample period is 24 hours, then there are 30 samples at the end of the month. The number of discard samples is two (7 percent of 30). In this case, days one, two, and three are in the initial period. During this period, target the lowest tier for each ISP at the start of each respective billing period and “walk up” the tiers until the current total amount of traffic is allocated across the links.

The middle period is after the initial period until the number of samples yet to be measured or collected is less than the number of discards. Using the same example as before, the middle period occurs from day four through day 28. During this period, set the target tier to the sustained utilization tier, which is the tier in which (discard +1) the highest sample so far measured falls.

The period after the middle period until the end of billing period is the last period. During this period, if you used links at the maximum link capacity for the remainder of the billing period and sustained utilization did not change, then set the target to the maximum allowable link utilization. Maximum link utilization is configurable where most likely values are 75 to 90 percent. Otherwise, set the target to sustained utilization tier.

During any sample period, if the cumulative usage is more than targeted cumulative usage, then bump up to the next tier for the remainder of sample period. If rollup is enabled, then replace the sample values with rollup values, and replace the number of samples with the number of rollups in the cost optimization algorithm.

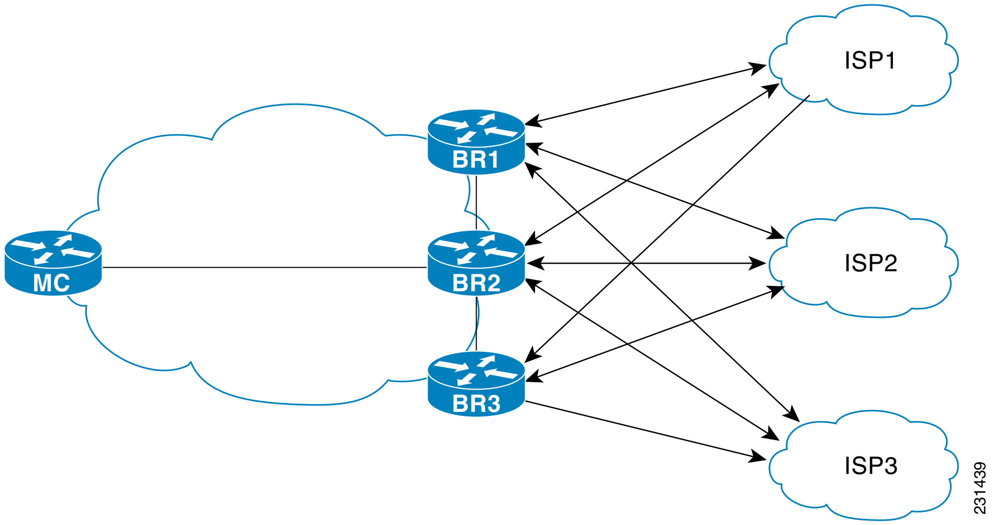
Performance Routing Link Grouping

In Cisco IOS Release 12.4(15)T the ability to define a group of exit links as a preferred set of links, or a fallback set of links for OER to use when optimizing traffic classes specified in an OER policy, was introduced. OER currently selects the best link for a traffic class based on the preferences specified in a policy and the traffic class performance—using parameters such as reachability, delay, loss, jitter or MOS—on a path out of the specified link. Bandwidth utilization, cost, and the range of links can also be considered in selecting the best link. Link grouping introduces a method of specifying preferred links for one or more traffic classes in an OER policy so that the traffic classes are routed through the best link from a list of preferred links, referred to as the primary link group. A fallback link group can also be specified in case there are no links in the primary group that satisfy the specified policy and performance requirements. If no primary group links are available, the traffic classes are routed through the best link from the fallback group. To identify the best exit, OER probes links from both the primary and fallback groups.

Primary and fallback link groups can be configured at the master controller and are identified using a unique name. Link groups provide a method of grouping links such as high bandwidth links to be used, for example, by video traffic, by configuring an OER policy to specify that the best link is to be selected from the link group that consists of only high bandwidth links. The traffic classes specified in a policy can be configured with only one primary link group and one fallback link group. The priority of a link group can vary between policies, a link group might be a primary link group for one policy, and a fallback link group for another policy.

See Figure 3 for an example of how to implement link grouping. Three link groups, ISP1, ISP2, and ISP3 represent different Internet Service Providers (ISPs) and all three ISPs have links to interfaces on the three border routers shown in Figure 3. ISP1 links are the most expensive links, but they have the best Service Level Agreement (SLA) guarantees. ISP3 links are best effort links, and these links are the cheapest links. ISP2 links are not as good as the ISP1 links, but the ISP2 links are more reliable than the ISP3 links. The cost of the ISP2 links is higher than the ISP3 links, but lower than ISP1 links. In this situation, each ISP is created as a link group and associated with an interface on each border router shown in Figure 3.

Figure 3 *Link Group Diagram*



Assuming four types of traffic class, video, voice, FTP, and data, each traffic class can be routed through a border router interface belonging to an appropriate link group. Video and voice traffic classes need the best links so the ISP1 link group is configured as the primary link group, with ISP2 as the fallback group. FTP traffic needs reliable links but the cost might be a factor so ISP2 is assigned as the primary group, and ISP3 is the fallback link group. Note that although ISP1 provides the most reliable links, it may be too expensive for file transfer traffic. For data traffic, ISP3 is a good choice as a primary link group, with ISP2 as the fallback group.

Spillover

Performance routing link groups can be used to support spillover. Spillover is when there are two paths through the network—traffic engineering (TE) tunnels, for example—to the same provider edge (PE) router, but the tunnels take different paths across the network and the traffic is sent through one tunnel until it reaches a traffic load threshold when it spills over to the second tunnel. Using OER link groups one tunnel is created as a primary link group and the second tunnel is the fallback link group. When the first tunnel goes out of policy, OER switches to the fallback tunnel link group, which provides the spillover capacity until the traffic load on the first tunnel drops below the threshold. The tunnels must be established before the OER link groups are configured.

OER Network Security Policies

The ability to configure network security policies either to prevent unauthorized use of the network or to mitigate attacks inside and outside the network was introduced in Cisco IOS Release 12.4(6)T and 12.2(33)SRB. You can configure OER to use black hole or sinkhole routing techniques to limit the impact of attacks against your network. Black hole routing refers to the process of forwarding packets to a null interface, meaning that the packets are dropped into a "black hole." Sinkhole routing directs packets to a next hop where the packets can be stored, analyzed, or dropped. Another term for sinkhole routing is honey-pot routing.

OER Policy Operational Options and Parameters

In addition to the specific types of OER policies, there are some OER policy operational parameters or options that can be configured. The operational parameters are timers and the operational options consist of different operational modes. For more details, see the following sections:

- OER Timers Parameters, page 10
- OER Mode Options, page 11

OER Timers Parameters

Three types of timers can be configured as OER policy operational parameters:

- Backoff Timer, page 10
- Holddown Timer, page 11
- Periodic Timer, page 11

Backoff Timer

The backoff timer is used to adjust the transition period that the master controller holds an out-of-policy traffic class entry. The master controller waits for the transition period before making an attempt to find an in-policy exit. A minimum, a maximum, and an optional step timer value can be configured.

Holddown Timer

The holddown timer is used to configure the traffic class entry route dampening timer to set the minimum period of time that a new exit must be used before an alternate exit can be selected. To prevent the traffic class entry from flapping because of rapid state changes, the master controller does not move the traffic class entry to a different exit even if it goes out-of-policy during the holddown timer period. OER does not implement policy changes while a traffic class entry is in the holddown state. A traffic class entry will remain in a holddown state for the default or configured time period. When the holddown timer expires, OER will select the best exit based on performance and policy configuration. However, an immediate route change will be triggered if the current exit for a traffic class entry becomes unreachable.

Periodic Timer

The periodic timer is used to find a better path for a traffic class entry, even if the traffic class entry is in-policy on the current exit. When the periodic timer expires, the master controller evaluates current exit links for the traffic class entry and, if a better exit exists based on the current measurements and priorities, the traffic class entry is moved to a new in-policy exit link.

When adjusting OER timers note that a newly configured timer setting will immediately replace the existing setting if the value of the new setting is less than the time remaining. If the value is greater than the time remaining, the new setting will be applied when the existing timer expires or is reset.

Note Overly aggressive timer settings can keep an exit link or traffic class entry in an out-of-policy state.

OER Mode Options

Three types of mode options can be configured as OER policy operational options:

- Mode Monitor, page 11
- Mode Route, page 11
- Mode Select-Exit, page 12

Mode Monitor

The mode monitor option enables the configuration of OER monitoring settings. Monitoring is defined here as the act of measurement performed periodically over a set interval of time where the measurements are compared against a threshold. OER measures the performance of traffic classes using active and passive monitoring techniques but it also measures, by default, the utilization of exit links. For more details about mode monitoring options, see the “Measuring the Traffic Class Performance and Link Utilization Using OER” module.

Mode Route

The mode route option specifies one of three OER route control policy settings. Mode route control enables OER to control routes automatically, mode route metric specifies OER route protocol-related settings, and mode route observe offers route control advice, but does not take any action. Observe mode monitoring is enabled by default when OER is enabled. In observe mode, the master controller monitors traffic classes and exit links based on default and user-defined policies and then reports the status of the network and the decisions that should be made but does not implement any changes. Observe mode is used to verify the effect of OER features before OER is actively deployed on your network. For more details about the mode route control and mode route metric options, see the “Using OER to Control the Traffic Classes and Verify the Network Performance” module.

Mode Select-Exit

The mode select-exit option enables the exit selection settings. The definition of an in-policy traffic class entry is that the measured performance metrics are do not exceed a default or configured threshold while the traffic class traffic is on the current path. In this situation, OER does not search for an alternate exit link because the current network path keeps the traffic class entry in-policy. This type of configuration would be activated by using the **mode select-link good** command which is the default if the **mode** command is not specified. There are other deployment scenarios, where OER selects the best performance path. This type of configuration can be activated by using the **mode select-link best** command. In this type of situation, OER measures alternate path performance metrics while the traffic class entry is in-policy on the current path. OER moves the current path if a better performance path is found. After the first selection of the best path, however, OER does not initiate another search unless the periodic timer is configured. When the periodic timer expires, the master controller evaluates current exit links for the traffic class entry and, if a better exit exists based on the current measurements and priorities, the traffic class entry is moved to a new in-policy exit link. Use the periodic timer with the **mode select-link best** command if you have a deployment scenario where you need OER to select the best performance path at any given time.

There is one further use of the mode select-exit option. If OER does not find an in-policy exit for a traffic class entry when the **mode select-link good** command is operational, OER transitions the traffic class entry to an uncontrolled state. If OER does not find an in-policy exit for a traffic class entry when the **mode select-link best** command is operational, OER selects the best of the OOP exit links for the traffic class entry.

OER Policy Application

OER policies can be applied to learned or configured traffic classes. OER policies can be applied on a global basis when the policy is configured directly under OER master controller configuration mode. All traffic classes inherit global policies. If, however, you want to apply a policy to a subset of the traffic classes, then a specific policy can be configured. A specific OER policy applies only to the specific traffic classes that match a prefix list or access list. Specific policies inherit global policies unless the same policy is overwritten by the specific policy. In Cisco IOS Release 12.4(6)T and earlier releases, OER policies applied only to prefixes, but in Cisco IOS Release 12.4(9)T, 12.2(33)SRB, and later releases, OER policies can apply to traffic classes that define an application traffic class and may include prefixes, protocols, port numbers, and DSCP values. To apply specific policies to learned or configured traffic classes, OER map configuration is used.

OER Map Configuration for OER Policies

An OER map may appear to be similar to a route map but there are significant differences. An OER map is designed to select learned or configured traffic classes using a match clause and then to apply OER policy configurations using a set clause. The OER map can be optionally configured with a sequence number like a route map, but only the OER map with the lowest sequence number is evaluated. The operation of an OER map differs from a route map at this point. There are two important distinctions:

- Only a single match clause may be configured for each sequence. An error message will be displayed on the console if you attempt to configure multiple match clauses for a single OER map sequence.
- An OER map is not configured with permit or deny statements. However, a permit or deny sequence can be configured for an IP traffic flow by configuring a permit or deny statement in an IP prefix list and then applying the prefix list to the OER map.

The OER map applies the configuration of the set clause after a successful match occurs. An OER set clause can be used to set policy parameters such as the backoff timer, packet delay, holddown timer, packet loss, mode settings, periodic timer, resolve settings, unreachable hosts, and traceroute reporting.

Policies applied by an OER map take effect immediately. The OER map configuration can be viewed in the output of the **show running-config** command. OER policy configuration can be viewed in the output of the **show oer master policy** command. These policies are applied only to traffic classes that match or pass through the OER map.

Policy Rules Configuration to Apply an OER Policy

The **policy-rules** OER master controller configuration command was introduced in Cisco IOS Release 12.3(11)T, 12.2(33)SRB, and later releases. This command allows you to select an OER map using a sequence number and apply the configuration under OER master controller configuration mode, providing an improved method to switch between predefined OER maps. Only one OER map is used at a time for policy configuration, but many OER maps can be defined. In Cisco IOS Release 12.3(8)T, only one OER map could be defined to apply a policy to traffic classes.

Priority Resolution for Multiple OER Policies

When configuring multiple policy criteria for a single traffic class entry, or a set of traffic classes, it is possible to have multiple overlapping policies. To resolve the potential conflict of which policy to run, OER uses its resolve function: a flexible mechanism that allows you to set the priority for an OER policy. Each policy is assigned a unique value, and the policy with the lowest value is selected as the highest priority policy. By default, OER assigns the highest priority to delay policies, followed by utilization policies. Assigning a priority value to any policy will override the default settings. To configure the policy conflict resolution, use the **resolve** command in OER master controller configuration mode, or the **set resolve** command in OER map configuration mode.

Variance Setting for OER Policy Conflict Resolution

When configuring OER resolve settings, you can also set an allowable variance for the defined policy. Variance configures the average delay, as a percentage, that all traffic classes for one exit, or the specific policy traffic classes for an exit, can vary from the defined policy value and still be considered equivalent. For example, if the delay on the best exit link (best exit in terms of delay) for a traffic class entry is 80 milliseconds (ms) and a 10 percent variance is configured, then any other exit links with a delay between 80 and 88 ms for the same traffic class entry are considered equivalent to the best exit link.

To illustrate how variance is used by OER consider three exit links with the following performance values for delay and jitter for a traffic class entry:

- Exit A—Delay is 80 ms, jitter is 3ms
- Exit B—Delay is 85 ms, jitter is 1ms
- Exit C—Delay is 100 ms, jitter is 5ms

The following OER policy conflict resolution is configured and applied to the traffic class entry:

```
delay priority 1 variance 10
jitter priority 2 variance 10
```

OER determines the best exit by looking at the policy with the lowest priority value, which in this example is the delay policy. Exit A has the lowest delay value, but Exit B has a delay value of 85 which is within a 10 percent variance of the delay value at Exit A. Exit A and Exit B can therefore be considered equal in terms of delay values. Exit C is now eliminated because the delay values are too high. The next priority policy is jitter, and Exit B has the lowest jitter value. OER will select Exit B as the only best exit for the traffic class entry because Exit A has a jitter value that is not within 10 percent variance of the Exit B jitter value.

Note: Variance cannot be configured for cost or range policies.

How to Configure and Apply OER Policies

This section contains the following optional tasks:

- Configuring and Applying an OER Policy to Learned Traffic Classes, page 14
- Configuring and Applying an OER Policy to Configured Traffic Classes, page 19
- Preventing OER Optimization of Learned Prefixes, page 26
- Configuring and Applying an OER Policy to Learned Inside Prefixes, page 29
- Configuring and Applying an OER Policy to Configured Inside Prefixes, page 32
- Configuring Policy Rules for OER Maps, page 35
- Configuring Multiple OER Policy Conflict Resolution, page 36
- Configuring an Exit Link Load Balancing OER Policy, page 38
- Implementing Performance Routing Link Groups, page 42
- Configuring OER Cost-Based Policies, page 48
- Configuring OER Network Security Policies, page 50
- Configuring OER Voice Traffic Optimization Using Active Probes, page 54

Configuring and Applying an OER Policy to Learned Traffic Classes

Perform this task at the master controller to configure and apply an OER policy to learned traffic classes. After configuring the router as an OER master controller using the **oer master** command, most of the commands in this task are all optional. Each step configures a performance policy that applies to learned traffic classes on a global basis. In this example, OER is configured to select the first in-policy exit.

Cisco IOS OER Timers Adjustments

When adjusting OER timers note that a newly configured timer setting will immediately replace the existing setting if the value of the new setting is less than the time remaining. If the value is greater than the time remaining, the new setting will be applied when the existing timer expires or is reset.

Note: Overly aggressive timer settings can keep an exit link or traffic class entry in an out-of-policy state.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **oer master**
4. **backoff** *min-timer max-timer* [*step-timer*]
5. **delay** {**relative** *percentage* | **threshold** *maximum*}

6. **holddown** *timer*
7. **loss** {**relative** *average* | **threshold** *maximum*}
8. **periodic** *timer*
9. **unreachable** {**relative** *average* | **threshold** *maximum*}
10. **mode select-exit** {**best** | **good**}
11. **end**
12. **show oer master policy** [*sequence-number* | *policy-name* | **default**]

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		- Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`oer master`**
	Example:
	`Router(config)# oer master`
		Enters OER master controller configuration mode.

Step 4	**`backoff`** *`min-timer max-timer`* `[`*`step-timer`*`]`
	Example:
	`Router(config-oer-mc)# backoff 400 4000 400`
		(Optional) Sets the backoff timer to adjust the time period for policy decisions.
		- The *min-timer* argument is used to set the minimum transition period in seconds.
		- The *max-timer* argument is used to set the maximum length of time OER holds an out-of-policy traffic class entry when there are no links that meet the policy requirements of the traffic class entry.
		- The *step-timer* argument allows you to optionally configure OER to add time each time the minimum timer expires until the maximum time limit has been reached.

	Command or Action	Purpose

Step 5	**delay** {**relative** *percentage* | **threshold** *maximum*}
	Example:
	`Router(config-oer-mc)# delay relative 80`
		(Optional) Sets the delay threshold as a relative percentage or as an absolute value.
		• The **relative** keyword is used to configure a relative delay percentage. The relative delay percentage is based on a comparison of short-term and long-term measurements.
		• The **threshold** keyword is used to configure the absolute maximum delay period in milliseconds.
		• If the configured delay threshold is exceeded, then the prefix is out-of-policy.
		• The example sets a delay threshold of 80 percent based on a relative average.

Step 6	**holddown** *timer* **Example:** `Router(config-oer-mc)# holddown 600`	(Optional) Configures the traffic class entry route dampening timer to set the minimum period of time that a new exit must be used before an alternate exit can be selected. • OER does not implement route changes while a traffic class entry is in the holddown state. • When the holddown timer expires, OER will select the best exit based on performance and policy configuration. • OER starts the process of finding an alternate path if the current exit for a traffic class entry becomes unreachable. • The example sets the traffic class entry route dampening timer to 600 seconds.

Step 7	**loss** {**relative** *average* | **threshold** *maximum*}
	Example:
	`Router(config-oer-mc)# loss relative 20`
		(Optional) Sets the relative or maximum packet loss limit that OER will permit for a traffic class entry.
		• The **relative** keyword sets a relative percentage of packet loss based on a comparison of short-term and long-term packet loss percentages.
		• The **threshold** keyword sets the absolute packet loss based on packets per million.
		• The example configures the master controller to search for a new exit link when the relative percentage of packet loss is equal to or greater than 20 percent.

	Command or Action	Purpose
Step 8	**periodic** *timer* **Example:** `Router(config-oer-mc)# periodic 300`	(Optional) Configures OER to periodically select the best exit link when the periodic timer expires. • When this command is enabled, the master controller will periodically evaluate and then make policy decisions for traffic classes. • The example sets the periodic timer to 300 seconds. When the timer expires, OER will select either the best exit or the first in-policy exit. **Note** The **mode select-exit** command is used to determine if OER selects the first in-policy exit or the best available exit when this timer expires. For more details, see the *“OER Mode Options” section on page 11.*

Step 9	**unreachable** {**relative** *average* | **threshold** *maximum*}
	Example:
	`Router(config-oer-mc)# unreachable relative 10`
		(Optional) Sets the maximum number of unreachable hosts.
		• This command is used to specify the relative percentage or the absolute maximum number of unreachable hosts, based on flows per million (fpm), that OER will permit for a traffic class entry. If the absolute number or relative percentage of unreachable hosts is greater than the user-defined or the default value, OER determines that the traffic class entry is OOP and searches for an alternate exit link.
		• The **relative** keyword is used to configure the relative percentage of unreachable hosts. The relative unreachable host percentage is based on a comparison of short-term and long-term measurements.
		• The **threshold** keyword is used to configure the absolute maximum number of unreachable hosts based on fpm.
		• The example configures OER to search for a new exit link for a traffic class entry when the relative percentage of unreachable hosts is equal to or greater than 10 percent.

Step 10	**mode select-exit** {**best** | **good**}}
	Example:
	`Router(config-oer-mc)# mode select-exit good`
		Enables the exit link selection based on performance or policy.
		• The **select-exit** keyword is used to configure the master controller to select either the best available exit when the **best** keyword is entered or the first in-policy exit when the **good** keyword is entered.
		Note Only the syntax that is applicable to this task is displayed. For more details, see the *Cisco IOS Optimized Edge Routing Command Reference*.

Step 11	**end**
	Example:
	`Router(config-oer-mc)# end`
		Exits OER master controller configuration mode and enters privileged EXEC mode.

	Command or Action	Purpose
Step 12	**`show oer master policy`** [*`sequence-number`* | *`policy-name`* | **`default`**] **Example:** `Router# show oer master policy`	Displays policy settings on an OER master controller. • The output of this command displays default policies and, optionally, policies configured with an OER map. • The *sequence-number* argument is used to display policy settings for the specified OER map sequence. • The *policy-name* argument is used to display policy settings for the specified OER policy map name. • The **default** keyword is used to display only the default policy settings. • The example displays the default policy settings and policy settings updated by the configuration in this task.

Examples

This example shows output from the **show oer master policy** command. Default policy settings are displayed except where the configuration in this task has overwritten specific policy settings.

```
Router# show oer master policy

Default Policy Settings:
  backoff 400 4000 400
  delay relative 80
  holddown 600
  periodic 300
  probe frequency 56
  mode route observe
  mode monitor both
  mode select-exit good
  loss relative 20
  unreachable relative 10
  resolve delay priority 11 variance 20
  resolve utilization priority 12 variance 20
 *tag 0
```

Configuring and Applying an OER Policy to Configured Traffic Classes

Perform this task at the master controller to configure and apply an OER policy to specified configured traffic classes. This task contains two targeted policies that work differently for different traffic class entries from the MTC list. The policies are configured using an OER map. This task contains both prefix list and access list configuration with different criteria in the set clauses. OER timers are also modified in this OER map configuration.

Note Policies applied in an OER map can override global policy configurations.

Cisco IOS OER Timers Adjustments

When adjusting OER timers note that a newly configured timer setting will immediately replace the existing setting if the value of the new setting is less than the time remaining. If the value is greater than the time remaining, the new setting will be applied when the existing timer expires or is reset.

Note Overly aggressive timer settings can keep an exit link or prefix in an out-of-policy state.

Prefix List Use with OER

IP prefix lists are used to manually select prefixes for OER monitoring and the prefix list syntax operates in a slightly different way with OER than in regular routing. The **ge** keyword is not used and the **le** keyword is used by OER to specify two types of prefixes: exact prefixes, and inclusive prefixes.

A master controller can monitor and control an exact prefix of any length including the default route. If an exact prefix is specified, OER monitors only the exact prefix.

A master controller can monitor and control an inclusive prefix using the **le** keyword and the *le-value* argument set to 32. OER monitors the configured prefix and any more specific prefixes (for example, configuring the 10.0.0.0/8 le 32 prefix would include the 10.1.0.0/16 and the 10.1.1.0/24 prefixes) over the same exit and records the information in the routing information base (RIB).

Note Use the inclusive prefix option with caution in a typical OER deployment because of the potential increase in the amount of prefixes being monitored and recorded.

Prerequisites

This task requires the master controller and border routers to be running Cisco IOS Release 12.4(6)T, 12.2(33)SRB, or later releases.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network/length* | **permit** *network/length*} [**le** *le-value*]
4. **ip access list** {**standard** | **extended**} *access-list-name*

5. [*sequence-number*] **permit udp** *source source-wildcard* [*operator* [*port*]] *destination destination-wildcard* [*operator* [*port*]] [**dscp** *dscp-value*] [
6. **exit**
7. **oer-map** *map-name sequence-number*
8. **match ip address prefix-list** *prefix-list-name*
9. **set backoff** *min-timer max-timer* [*step-timer*]
10. **set delay** {**relative** *percentage* | **threshold** *maximum*}
11. **set loss** {**relative** *average* | **threshold** *maximum*}
12. **set periodic** *timer*
13. **set unreachable** {**relative** *average* | threshold *maximum*}
14. **exit**
15. **oer-map** *map-name sequence-number*
16. **match ip address** {**access-list** *access-list-name* | **prefix-list** *prefix-list-name*}
17. **set active probe** *probe-type ip-address* [**target-port** *number*] [**codec** *codec-name*]
18. **set probe frequency** *seconds*
19. **set jitter threshold** *maximum*
20. **set mos threshold** *minimum* **percent** *percent*
21. **set mode select-exit** {**best** | **good**}
22. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network***/***length* | **permit** *network***/***length*} [**le** *le-value*] **Example:** `Router(config)# ip prefix-list OER seq 10 permit 10.4.9.0/24`	Creates an IP prefix list. • IP prefix lists are used to manually select prefixes for monitoring by the master controller. • A master controller can monitor and control an exact prefix of any length including the default route. If an exact prefix is specified, OER monitors only the exact prefix. • A master controller can monitor and control an inclusive prefix using the **le** keyword set to 32. OER monitors the configured prefix and any more specific prefixes (for example, configuring the 10.0.0.0/8 le 32 prefix would include the 10.1.0.0/16 and the 10.1.1.0/24 prefixes) over the same exit and records the information in the routing information base (RIB). • The prefixes specified in the IP prefix list are imported into the OER map with the **match ip address** (OER) command. • The example creates an exact IP prefix list that permits prefixes only from the 10.4.9.0/24 subnet. **Note** Only the syntax applicable to OER is shown. For more details, see the *Cisco IOS IP Routing Protocols Command Reference*.
Step 4	**ip access-list** {**standard** | **extended**} *access-list-name* **Example:** `Router(config)# ip access-list extended VOICE_ACCESS_LIST`	Defines an IP access list by name. • OER supports only named access lists. • The example creates an extended IP access list named VOICE_ACCESS_LIST.
Step 5	[*sequence-number*] **permit udp** *source source-wildcard* [*operator* [*port*]] *destination destination-wildcard* [*operator* [*port*]] [**dscp** *dscp-value*] **Example:** `Router(config-ext-nacl)# permit udp any range 16384 32767 10.20.20.0 0.0.0.15 range 16384 32767`	Sets conditions to allow a packet to pass a named IP access list. • The example is configured to identify all UDP traffic ranging from a destination port number of 16384 to 32767 from any source to a destination prefix of 10.20.20.0/24. This specific UDP traffic is to be optimized. **Note** Only the syntax applicable to this task is shown. For more details, see the *Cisco IOS IP Application Services Command Reference*.
Step 6	**exit** **Example:** `Router(config-ext-nacl)# exit`	Exits extended access list configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 7	**oer-map** *map-name sequence-number* **Example:** `Router(config)# oer-map FINANCE 10`	Enters OER map configuration mode to configure an OER map to apply policies to selected IP prefixes. • *Only one match clause can be configured for each OER map sequence.* • Permit sequences are first defined in an IP prefix list and then applied with the **match ip address** (OER) command in Step 8. • The example creates an OER map named FINANCE.

Step 8	**match ip address** {**access-list** *access-list-name* | **prefix-list** *prefix-list-name*}
	Example:
	`Router(config-oer-map)# match ip address prefix-list OER`
		References an extended IP access list or IP prefix list as match criteria in an OER map.
		• Only a single match clause can be configured for each OER map sequence.
		• The example configures the prefix list named OER as match criteria in an OER map.

Step 9	**set backoff** *min-timer max-timer* [*step-timer*] **Example:** `Router(config-oer-map)# set backoff 400 4000 400`	Creates a set clause entry to configure the backoff timer to adjust the time period for traffic class entry policy decisions. • The *min-timer* argument is used to set the minimum transition period in seconds. • The *max-timer* argument is used to set the maximum length of time OER holds an out-of-policy traffic class entry when there are no OER controlled in-policy traffic classes. • The *step-timer* argument allows you to optionally configure OER to add time each time the minimum timer expires until the maximum time limit has been reached. • The example creates a set clause to configure the minimum timer to 400 seconds, the maximum timer to 4000 seconds, and the step timer to 400 seconds for traffic that is matched in the same OER map sequence.

Step 10	**set delay** {**relative** *percentage* | **threshold** *maximum*}
	Example:
	`Router(config-oer-map)# set delay threshold 2000`
		Creates a set clause entry to configure the delay threshold.
		• The delay threshold can be configured as a relative percentage or as an absolute value for match criteria.
		• The **relative** keyword is used to configure a relative delay percentage. The relative delay percentage is based on a comparison of short-term and long-term measurements.
		• The **threshold** keyword is used to configure the absolute maximum delay period in milliseconds.
		• The example creates a set clause that sets the absolute maximum delay threshold to 2000 milliseconds for traffic that is matched in the same OER map sequence.

	Command or Action	Purpose

Step 11	**`set loss`** `{`**`relative`** *`average`* `|` **`threshold`** *`maximum`*`}`
	Example:
	`Router(config-oer-map)# set loss relative 20`
		Creates a set clause entry to configure the relative or maximum packet loss limit that the master controller will permit for an exit link.
		• This command is used within an OER map to configure the relative percentage or maximum number of packets that OER will permit to be lost during transmission on an exit link. If packet loss is greater than the user-defined or the default value, the master controller determines that the exit link is out-of-policy.
		• The **relative** keyword is used to configure the relative packet loss percentage. The relative packet loss percentage is based on a comparison of short-term and long-term packet loss.
		• The **threshold** keyword is used to configure the absolute maximum packet loss. The maximum value is based on the actual number of packets per million that have been lost.
		• The example creates a set clause that configures the relative percentage of acceptable packet loss to less than 20 percent for traffic that is matched in the same OER map sequence.

Step 12	**`set periodic`** *`timer`*
	Example:
	`Router(config-oer-map)# set periodic 300`
		Creates a set clause entry to configure the time period for the periodic timer.
		• When this command is enabled, the master controller will periodically evaluate and then make policy decisions for traffic classes, even if they are currently in-policy.
		• The **set mode select-exit** command in Step 21 is used to determine if OER selects the first in-policy exit or the best available exit when this timer expires.
		• The example creates a set clause that configures the periodic timer to 300 seconds for traffic that is matched in the same OER map sequence.

	Command or Action	Purpose

Step 13	**set unreachable** {**relative** *average* | **threshold** *maximum*}
	Example:
	`Router(config-oer-map)# set unreachable relative 10`
		Creates a set clause entry to configure the maximum number of unreachable hosts.
		• This command is used to specify the relative percentage or the absolute maximum number of unreachable hosts, based on flows per million (fpm), that OER will permit for a traffic class entry. If the absolute number or relative percentage of unreachable hosts is greater than the user-defined or the default value, OER determines that the traffic class entry is OOP and searches for an alternate exit link.
		• The **relative** keyword is used to configure the relative percentage of unreachable hosts. The relative unreachable host percentage is based on a comparison of short-term and long-term measurements.
		• The **threshold** keyword is used to configure the absolute maximum number of unreachable hosts based on fpm.
		• The example creates a set clause entry that configures the master controller to search for a new exit link for a traffic class entry when the relative percentage of unreachable hosts is equal to or greater than 10 percent for traffic learned based on highest delay.

Step 14	**exit** **Example:** `Router(config-oer-map)# exit`	(Optional) Exits OER map configuration mode and returns to global configuration mode.
Step 15	**oer-map** *map-name sequence-number* **Example:** `Router(config)# oer-map VOICE_MAP 10`	Enters OER map configuration mode to configure an OER map to apply policies to selected IP traffic classes. • *Only one match clause can be configured for each OER map sequence.* • Deny sequences are first defined in an IP prefix list and then applied with the **match ip address** (OER) command in Step 16. • The example creates an OER map named VOICE_MAP.

Step 16	**match ip address** {**access-list** *access-list-name* | **prefix-list** *prefix-list-name*}
	Example:
	`Router(config-oer-map)# match ip address access-list VOICE_ACCESS_LIST`
		References an extended IP access list or IP prefix list as match criteria in an OER map.
		• Only a single match clause can be configured for each OER map sequence.
		• The example configures the IP access list named VOICE_ACCESS_LIST as match criteria in an OER map.

	Command or Action	Purpose
Step 17	**set active-probe** *probe-type ip-address* [**target-port** *number*] [**codec** *codec-name*] **Example:** `Router(config-oer-map)# set active-probe jitter 10.20.22.1 target-port 2000 codec g729a`	Creates a set clause entry to assign a target prefix for an active probe. • The **echo** keyword is used to specify the target IP address of a prefix to actively monitor using Internet Control Message Protocol (ICMP) echo (ping) messages. • The **jitter** keyword is used to specify the target IP address of a prefix to actively monitor using jitter messages. • The **tcp-conn** keyword is used to specify the target IP address of a prefix to actively monitor using Internet Control Message Protocol (ICMP) echo (ping) messages. • The **udp-echo** keyword is used to specify the target IP address of a prefix to actively monitor using Internet Control Message Protocol (ICMP) echo (ping) messages. • The example creates a set clause entry to specify the target IP address of a prefix and a specific port number to actively monitor using jitter.
Step 18	**set probe frequency** *seconds* **Example:** `Router(config-oer-map)# set probe frequency 10`	Creates a set clause entry to set the frequency of the OER active probe. • The *seconds* argument is used to set the time, in seconds, between the active probe monitoring of the specified IP prefixes. • The example creates a set clause to set the active probe frequency to 10 seconds.
Step 19	**set jitter threshold** *maximum* **Example:** `Router(config-oer-map)# set jitter threshold 20`	Creates a set clause entry to configure the jitter threshold value. • The **threshold** keyword is used to configure the maximum jitter value, in milliseconds. • The example creates a set clause that sets the jitter threshold value to 20 for traffic that is matched in the same OER map sequence.

	Command or Action	Purpose

Step 20	**set mos** {**threshold** *minimum* **percent** *percent*}
	Example:
	`Router(config-oer-map)# set mos threshold 4.0 percent 30`
		Creates a set clause entry to configure the MOS threshold and percentage values used to decide whether an alternate exit is be selected.
		• The **threshold** keyword is used to configure the minimum MOS value.
		• The **percent** keyword is used to configure the percentage of MOS values that are below the MOS threshold.
		• OER calculates the percentage of MOS values below the MOS threshold that are recorded in a five-minute period. If the percentage value exceeds the configured percent value or the default value, the master controller searches for alternate exit links.
		• The example creates a set clause that sets the threshold MOS value to 4.0 and the percent value to 30 percent for traffic that is matched in the same OER map sequence.

Step 21	**set mode select-exit** {**best** | **good**}
	Example:
	`Router(config-oer-map)# set mode select-exit best`
		Creates a set clause entry to configure monitoring, control, or exit selection settings for matched traffic.
		• The **select-exit** keyword is used to configure the master controller to select either the best available exit when the **best** keyword is entered or the first in-policy exit when the **good** keyword is entered.

Step 22	**end**
	Example:
	`Router(config-oer-map)# end`
		Exits OER map configuration mode and enters privileged EXEC mode.

Preventing OER Optimization of Learned Prefixes

Perform this task at the master controller to configure and apply an OER policy to prevent OER from attempting to optimize specified learned prefixes. This task is useful when you know a few prefixes that you want to exclude from the OER optimization, but these prefixes will be learned automatically by OER. In this task, an IP prefix list is configured with two entries for different prefixes that are not to be optimized. An OER map is configured with two entries in a sequence that will prevent OER from optimizing the prefixes specified in the prefix list, although the prefixes may be learned. If the sequence numbers of the OER map entries are reversed, OER will learn and attempt to optimize the prefixes.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network/length* | **permit** *network/length*} [**le** *le-value*]
4. **ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network/length* | **permit** *network/length*} [**le** *le-value*]
5. **oer-map** *map-name sequence-number*

6. **match ip address prefix-list** *prefix-list-name*
7. **oer-map** *map-name sequence-number*
8. **match oer learn** {**delay** | **inside** | **throughput**}
9. end

DETAILED STEPS

	Command or Action	Purpose

Step 1	`enable`
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	`configure terminal`
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network***/***length* | **permit** *network***/***length*} [**le** *le-value*] **Example:** `Router(config)# ip prefix-list DENY_LIST deny 10.1.1.0/24`	Creates an IP prefix list. • IP prefix lists are used to manually deny or permit prefixes for monitoring by the master controller. • The prefixes specified in the IP prefix list are imported into the OER map with the **match ip address** (OER) command. • The example creates an IP prefix list with an entry that denies prefixes only from the 10.1.1.0/24 subnet. **Note** Only the syntax applicable to OER is shown. For more details, see the *Cisco IOS IP Routing Protocols Command Reference*, Release 12.4T.
Step 4	**ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network***/***length* | **permit** *network***/***length*} [**le** *le-value*] **Example:** `Router(config)# ip prefix-list DENY_LIST deny 172.20.1.0/24`	Creates an IP prefix list. • IP prefix lists are used to manually deny or permit prefixes for monitoring by the master controller. • The prefixes specified in the IP prefix list are imported into the OER map with the **match ip address** (OER) command. • The example creates an IP prefix entry that denies prefixes only from the 172.20.1.0/24 subnet. **Note** Only the syntax applicable to OER is shown. For more details, see the *Cisco IOS IP Routing Protocols Command Reference*, Release 12.4T.

	Command or Action	Purpose
Step 5	**oer-map** *map-name sequence-number* **Example:** `Router(config)# oer-map DENY_MAP 10`	Enters OER map configuration mode to configure an OER map to apply policies to selected IP prefixes. • *Only one match clause can be configured for each OER map sequence.* • Deny sequences are first defined in an IP prefix list and then applied with the **match ip address** (OER) command in Step 6. • The example creates an OER map named DENY_MAP with a sequence number of 10.

Step 6	**match ip address** {**access-list** *access-list-name* | **prefix-list** *prefix-list-name*}
	Example:
	`Router(config-oer-map)# match ip address prefix-list DENY_LIST`
		References an extended IP access list or IP prefix list as match criteria in an OER map.
		• Only a single match clause can be configured for each OER map sequence.
		• The example configures the prefix list named OER as match criteria in an OER map.

Step 7	**exit** **Example:** `Router(config-oer-map)# exit`	Exits OER map configuration mode and returns to global configuration mode.
Step 8	**oer-map** *map-name sequence-number* **Example:** `Router(config)# oer-map DENY_MAP 20`	Enters an OER map entry. • *Only one match clause can be configured for each OER map sequence.* • Deny sequences are first defined in an IP prefix list and then applied with the **match ip address** (OER) command in Step 9. • The example creates an OER map entry for the OER map named DENY_MAP with a sequence number of 20.

Step 9	**match oer learn** {**delay** | **inside** | **throughput**}
	Example:
	`Router(config-oer-map)# match oer learn throughput`
		Creates a match clause entry in an OER map to match OER learned prefixes.
		• OER can be configured to learn traffic classes that are inside prefixes or prefixes based on highest delay, or highest outbound throughput.
		• Only a single match clause can be configured for each OER map sequence.
		• The example creates a match clause entry that matches traffic classes that are learned on the basis of the highest throughput.

Step 10	**end**
	Example:
	`Router(config-oer-map)# end`
		(Optional) Exits OER map configuration mode and returns to privileged EXEC mode.

Configuring and Applying an OER Policy to Learned Inside Prefixes

Perform this task to apply a policy to learned inside prefix traffic class entries from the MTC list. Support for optimizing inside prefixes was introduced in Cisco IOS Release 12.4(9)T and 12.2(33)SRB. The policy is configured using an OER map and contains some set clauses.

Note Policies applied in an OER map do not override global policy configurations.

OER Inside Prefixes

An OER inside prefix is defined as a public IP prefix assigned to a company. An OER outside prefix is defined as a public IP prefix assigned outside the company. Companies advertise the inside prefixes over the Internet using an Internet service provider (ISP) and receive advertisements for outside prefixes from an ISP.

Prerequisites

This task requires the master controller and border routers to be running Cisco IOS Release 12.4(9)T, 12.2(33)SRB, or later releases.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **oer-map** *map-name sequence-number*
4. **match oer learn** {**delay** | **inside** | **throughput**}
5. **set delay** {**relative** *percentage* | **threshold** *maximum*}
6. **set loss** {**relative** *average* | **threshold** *maximum*}
7. **set unreachable** {**relative** *average* | threshold *maximum*}
8. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`oer-map`** *`map-name sequence-number`*
	Example:
	`Router(config)# oer-map INSIDE_LEARN 10`
		Enters OER map configuration mode to configure an OER map to apply policies to selected IP prefixes.
		• *Only one match clause can be configured for each OER map sequence.*
		• Deny sequences are first defined in an IP prefix list and then applied with a **match** command.
		• The example creates an OER map named INSIDE_LEARN.

Step 4	**`match oer learn`** {**`delay`** | **`inside`** | **`throughput`**}
	Example:
	`Router(config-oer-map)# match oer learn inside`
		Creates a match clause entry in an OER map to match OER learned prefixes.
		• Prefixes can be configured to learn prefixes that are inside prefixes or prefixes based on lowest delay, or highest outbound throughput.
		• Only a single match clause can be configured for each OER map sequence.
		• The example creates a match clause entry that matches traffic learned using inside prefixes.

Step 5	**`set delay`** {**`relative`** *`percentage`* | **`threshold`** *`maximum`*}
	Example:
	`Router(config-oer-map)# set delay threshold 2000`
		Creates a set clause entry to configure the delay threshold.
		• The delay threshold can be configured as a relative percentage or as an absolute value for match criteria.
		• The **relative** keyword is used to configure a relative delay percentage. The relative delay percentage is based on a comparison of short-term and long-term measurements.
		• The **threshold** keyword is used to configure the absolute maximum delay period in milliseconds.
		• The example creates a set clause that sets the absolute maximum delay threshold to 2000 milliseconds for traffic that is matched in the same OER map sequence.

	Command or Action	Purpose

Step 6	**set loss** {**relative** *average* | **threshold** *maximum*}
	Example:
	`Router(config-oer-map)# set loss relative 20`
		Creates a set clause entry to configure the relative or maximum packet loss limit that the master controller will permit for an exit link.
		• This command is used to configure an OER map to configure the relative percentage or maximum number of packets that OER will permit to be lost during transmission on an exit link. If packet loss is greater than the user-defined or the default value, the master controller determines that the exit link is out-of-policy.
		• The **relative** keyword is used to configure the relative packet loss percentage. The relative packet loss percentage is based on a comparison of short-term and long-term packet loss.
		• The **threshold** keyword is used to configure the absolute maximum packet loss. The maximum value is based on the actual number of packets per million that have been lost.
		• The example creates a set clause that configures the relative percentage of acceptable packet loss to less than 20 percent for traffic that is matched in the same OER map sequence.

Step 7	**set unreachable** {**relative** *average* | **threshold** *maximum*}
	Example:
	`Router(config-oer-map)# set unreachable relative 10`
		Creates a set clause entry to configure the maximum number of unreachable hosts.
		• This command is used to specify the relative percentage or the absolute maximum number of unreachable hosts, based on flows per million (fpm), that OER will permit for a traffic class entry. If the absolute number or relative percentage of unreachable hosts is greater than the user-defined or the default value, OER determines that the traffic class entry is OOP and searches for an alternate exit link.
		• The **relative** keyword is used to configure the relative percentage of unreachable hosts. The relative unreachable host percentage is based on a comparison of short-term and long-term measurements.
		• The **threshold** keyword is used to configure the absolute maximum number of unreachable hosts based on fpm.
		• The example creates a set clause entry that configures the master controller to search for a new exit link for a traffic class entry when the relative percentage of unreachable hosts is equal to or greater than 10 percent for traffic learned based on highest delay.

Step 8	**end**
	Example:
	`Router(config-oer-map)# end`
		(Optional) Exits OER map configuration mode and returns to privileged EXEC mode.

Configuring and Applying an OER Policy to Configured Inside Prefixes

Perform this task to apply a policy to configured inside prefix traffic class entries from the MTC list. Support for optimizing inside prefixes was introduced in Cisco IOS Release 12.4(9)T and 12.2(33)SRB. The policies are configured using an OER map. This task contains prefix list configuration with different criteria in the set clauses.

Note Policies applied in an OER map do not override global policy configurations.

OER Inside Prefixes

An OER inside prefix is defined as a public IP prefix assigned to a company. An OER outside prefix is defined as a public IP prefix assigned outside the company. Companies advertise the inside prefixes over the Internet using an Internet service provider (ISP) and receive advertisements for outside prefixes from an ISP.

Prerequisites

This task requires the master controller and border routers to be running Cisco IOS Release 12.4(9)T, 12.2(33)SRB, or later releases.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **oer-map** *map-name sequence-number*
4. **match ip address** {**access-list** *access-list-name* | **prefix-list** *prefix-list-name* [**inside**]}
5. **set delay** {**relative** *percentage* | **threshold** *maximum*}
6. **set loss** {**relative** *average* | **threshold** *maximum*}
7. **set unreachable** {**relative** *average* | threshold *maximum*}
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**oer-map map-name** *sequence-number* **Example:** `Router(config)# oer-map INSIDE_CONFIGURE 10`	Enters OER map configuration mode to create or configure an OER map. • *OER map operation is similar to that of route maps.* • *Only a single match clause can be configured for each OER map sequence.* • Common and deny sequences should be applied to lowest oer-map sequence for best performance. • The example creates an OER map named INSIDE_CONFIGURE.
Step 4	**match ip address** {**access-list** *access-list-name* | **prefix-list** *prefix-list-name* [**inside**] **Example:** `Router(config-oer-map)# match ip address prefix-list INSIDE_PREFIXES inside`	References an extended IP access list or IP prefix list as match criteria in an OER map. • Use the **inside** keyword to specify inside prefixes to support OER BGP inbound optimization that supports best entrance selection for traffic that originates from prefixes outside an autonomous system destined for prefixes inside the autonomous system. • The example creates a match clause entry using the prefix list INSIDE_PREFIXES that specifies inside prefixes.

Step 5	**set delay** {**relative** *percentage* | **threshold** *maximum*}
	Example:
	`Router(config-oer-map)# set delay threshold 2000`
		Creates a set clause entry to configure the delay threshold.
		• The delay threshold can be configured as a relative percentage or as an absolute value for match criteria.
		• The **relative** keyword is used to configure a relative delay percentage. The relative delay percentage is based on a comparison of short-term and long-term measurements.
		• The **threshold** keyword is used to configure the absolute maximum delay period in milliseconds.
		• The example creates a set clause that sets the absolute maximum delay threshold to 2000 milliseconds for traffic that is matched in the same OER map sequence.

	Command or Action	Purpose

Step 6	**set loss** {**relative** *average* | **threshold** *maximum*}
	Example:
	`Router(config-oer-map)# set loss relative 20`
		Creates a set clause entry to configure the relative or maximum packet loss limit that the master controller will permit for an exit link.
		• This command is used to configure an OER map to configure the relative percentage or maximum number of packets that OER will permit to be lost during transmission on an exit link. If packet loss is greater than the user-defined or the default value, the master controller determines that the exit link is out-of-policy.
		• The **relative** keyword is used to configure the relative packet loss percentage. The relative packet loss percentage is based on a comparison of short-term and long-term packet loss.
		• The **threshold** keyword is used to configure the absolute maximum packet loss. The maximum value is based on the actual number of packets per million that have been lost.
		• The example creates a set clause that configures the relative percentage of acceptable packet loss to less than 20 percent for traffic that is matched in the same OER map sequence.

Step 7	**set unreachable** {**relative** *average* | **threshold** *maximum*}
	Example:
	`Router(config-oer-map)# set unreachable relative 10`
		Creates a set clause entry to configure the maximum number of unreachable hosts.
		• This command is used to specify the relative percentage or the absolute maximum number of unreachable hosts, based on flows per million (fpm), that OER will permit for a traffic class entry. If the absolute number or relative percentage of unreachable hosts is greater than the user-defined or the default value, OER determines that the traffic class entry is OOP and searches for an alternate exit link.
		• The **relative** keyword is used to configure the relative percentage of unreachable hosts. The relative unreachable host percentage is based on a comparison of short-term and long-term measurements.
		• The **threshold** keyword is used to configure the absolute maximum number of unreachable hosts based on fpm.
		• The example creates a set clause entry that configures the master controller to search for a new exit link for a traffic class entry when the relative percentage of unreachable hosts is equal to or greater than 10 percent for traffic learned based on highest delay.

Step 8	**end**
	Example:
	`Router(config-oer-map)# end`
		Exits OER map configuration mode and returns to privileged EXEC mode.

Configuring Policy Rules for OER Maps

Perform this task to select an OER map and apply the configuration under OER master controller configuration mode. The **policy-rules** OER master controller configuration command was introduced in Cisco IOS Release 12.3(11)T, and this command provides an improved method to switch between predefined OER maps.

Prerequisites

- At least one OER map must be configured before you can enable policy-rule support.
- This task requires the master controller and border routers to be running Cisco IOS Release 12.3(11)T, 12.2(33)SRB, or later release.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **oer master**
4. **policy-rules** *map-name*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**oer master** **Example:** `Router(config)# oer master`	Enters OER master controller configuration mode to configure global prefix and exit link policies.

	Command or Action	Purpose
Step 4	`policy-rules` *map-name* **Example:** `Router(config-oer-mc)# policy-rules RED`	Applies a configuration from an OER map to a master controller configuration in OER master controller configuration mode. • Reentering this command with a new OER map name will immediately overwrite the previous configuration. This behavior is designed to allow you to quickly select and switch between predefined OER maps. • The example applies the configuration from the OER map named RED.

Step 5	`end`
	Example:
	`Router(config-oer-mc)# end`
		Exits OER master controller configuration mode and enters privileged EXEC mode.

Configuring Multiple OER Policy Conflict Resolution

Perform this task to use the OER resolve function to assign a priority to an OER policy to avoid any conflict over which policy to run first. Each policy is assigned a unique value, and the policy with the highest value is selected as the highest priority. By default, a delay policy has the highest priority and a traffic load (utilization) policy has the second highest priority. Assigning a priority value to any policy will override default settings.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **oer master**
4. **resolve** {**cost priority** *value* | **delay priority** *value* **variance** *percentage* | **loss priority** *value* **variance** *percentage* | **range priority** *value* | **utilization priority** *value* **variance** *percentage*}
5. Repeat Step 4 to assign a priority for each required OER policy.
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**oer master** **Example:** `Router(config)# oer master`	Enters OER master controller configuration mode.

Step 4	**resolve** {**cost priority** *value* | **delay priority** *value* **variance** *percentage* | **loss priority** *value* **variance** *percentage* | **range priority** *value* | **utilization priority** *value* **variance** *percentage*}
	Example:
	`Router(config-oer-mc)# resolve loss priority 2 variance 10`
		Sets policy priority or resolves policy conflicts.
		• This command is used to set priority when multiple policies are configured for the same prefix. When this command is configured, the policy with the highest priority will be selected to determine the policy decision.
		• The **priority** keyword is used to specify the priority value. Setting the number 1 assigns the highest priority to a policy. Setting the number 10 assigns the lowest priority.
		• Each policy must be assigned a different priority number.
		• The **variance** keyword is used to set an allowable variance for a user-defined policy. This keyword configures the allowable percentage that an exit link or prefix can vary from the user-defined policy value and still be considered equivalent.
		• The example sets the priority for loss policies to 2 with a 10 percent variance.
		Note Variance cannot be configured for range or cost policies.
		Note Support for **jitter** and **mos** policies was introduced in Cisco IOS Release 12.4(6)T and 12.2(33)SRB. For more details, see the *Cisco IOS Optimized Edge Routing Command Reference*.

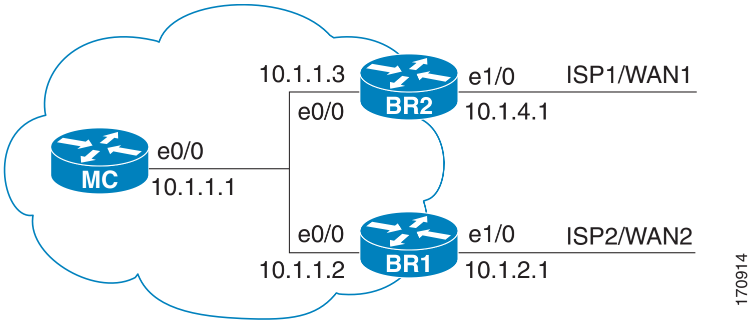
Step 5	Repeat Step 4 to assign a priority for each required OER policy.	—

Step 6	**end**
	Example:
	`Router(config-oer-mc)# end`
		Exits OER master controller configuration mode, and enters privileged EXEC mode.

Configuring an Exit Link Load Balancing OER Policy

Perform this task at the master controller to configure a load balancing policy for traffic class flows over the border router exit links. In this example, range and exit utilization policies are given priority when OER chooses the best exit selection for traffic class flows. Best route selection for performance policies is disabled. The external Ethernet interfaces on border router 1 and border router 2—BR1 and BR2 in Figure 4—are both configured with a maximum utilization threshold of 70 percent and a range of utilization between the two exit links is set to 30 percent. After an external interface is configured for the border routers, OER automatically monitors the utilization of external links on a border router every 5 minutes. The utilization is reported back to the master controller and, if the utilization exceeds 70 percent, OER selects another exit link for traffic class flows on that link. To complete the load balancing, the utilization range between the two exit links must not be greater than 30 percent, otherwise OER will move some of the traffic classes from one exit link to another to balance the traffic load between the two exit links.

Figure 4 Network diagram for OER Exit Link Load Balancing



Traffic can also be load balanced over entrance links, for more details see the “Using OER to Control Traffic Classes and Verify the Network Performance” module.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **oer master**
4. **max-range-utilization percent** *maximum*
5. **mode select-exit** {**best** | **good**}
6. **resolve range priority** *value*
7. **resolve utilization priority** *value* **variance** *percentage*
8. **no resolve delay**
9. **no resolve loss**
10. **border** *ip-address* [**key-chain** *key-chain-name*]
11. **interface** *type number* **external**
12. **max-xmit-utilization** {**absolute** *kbps* | **percentage** *value*}
13. **exit**
14. **exit**

15. Repeat Step 10 through Step 14 to set a utilization threshold for each external link.
16. **keepalive** *timer*
17. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`oer master`**
	Example:
	`Router(config)# oer master`
		Enters OER master controller configuration mode to configure a router as a master controller and to configure global operations and policies.

Step 4	**`max-range-utilization percent`** *`maximum`*
	Example:
	`Router(config-oer-mc)# max-range-utilization percent 30`
		Sets the maximum utilization range for all OER-managed exit link.s.
		• Use the **percent** keyword and *maximum* argument to specify the maximum utilization range between all the exit links.
		• In this example, the utilization range between all the exit links on the border routers must be within 30 percent.

Step 5	**`mode select-exit`** `{`**`best`** `|` **`good`**`}`
	Example:
	`Router(config-oer-mc)# mode select-exit best`
		Creates a set clause entry to configure exit selection settings.
		• Use the **select-exit** keyword to configure the master controller to select either the best available exit when the **best** keyword is entered or the first in-policy exit when the **good** keyword is entered.
		• In this example, OER will select the best available exit.
		Note Only the syntax relevant to this task is displayed. For more details, see the *Cisco IOS Optimized Edge Routing Command Reference*.

	Command or Action	Purpose
Step 6	**resolve range priority** *value* **Example:** `Router(config-oer-mc)# resolve range priority 1`	Sets policy priority or resolves policy conflicts. • This command is used to set the priorities when multiple policies are configured for the same prefix. When this command is configured, the policy with the highest priority will be selected to determine the policy decision. • The **priority** keyword is used to specify the priority value. Setting the number 1 assigns the highest priority to a policy. Setting the number 10 assigns the lowest priority. • Each policy must be assigned a different priority number. • In this example, the priority for range policies is set to 1. **Note** Only the syntax relevant to this task is displayed. For more details, see the *Cisco IOS Optimized Edge Routing Command Reference*.
Step 7	**resolve utilization priority** *value* **variance** *percentage* **Example:** `Router(config-oer-mc)# resolve utilization priority 2 variance 25`	Sets policy priority or resolves policy conflicts. • This command is used to set the priorities when multiple policies are configured for the same prefix. When this command is configured, the policy with the highest priority will be selected to determine the policy decision. • The **priority** keyword is used to specify the priority value. Setting the number 1 assigns the highest priority to a policy. Setting the number 10 assigns the lowest priority. • Each policy must be assigned a different priority number. • The **variance** keyword is used to set an allowable variance for a user-defined policy. This keyword configures the allowable percentage that an exit link or prefix can vary from the user-defined policy value and still be considered equivalent. • In this example, the priority for utilization policies is set to 2 with a 25 percent variance. **Note** Only the syntax relevant to this task is displayed. For more details, see the *Cisco IOS Optimized Edge Routing Command Reference*.
Step 8	**no resolve delay** **Example:** `Router(config-oer-mc)# no resolve delay`	Disables any priority for delay performance policies. **Note** Only the syntax relevant to this task is displayed. For more details, see the *Cisco IOS Optimized Edge Routing Command Reference*.

	Command or Action	Purpose
Step 9	**no resolve loss** **Example:** `Router(config-oer-mc)# no resolve loss`	Disables any priority for loss performance policies. **Note** Only the syntax relevant to this task is displayed. For more details, see the *Cisco IOS Optimized Edge Routing Command Reference*.
Step 10	**border** *ip-address* [**key-chain** *key-chain-name*] **Example:** `Router(config-oer-mc)# border 10.1.1.2 key-chain border1_OER`	Enters OER-managed border router configuration mode to establish communication with a border router. • An IP address is configured to identify the border router. • At least one border router must be specified to create an OER-managed network. A maximum of ten border routers can be controlled by a single master controller. • The value for the *key-chain-name* argument must match a valid the key-chain name configured on the border router. **Note** The **key-chain** keyword and *key-chain-name* argument must be entered when a border router is initially configured. However, this keyword is optional when reconfiguring an existing border router.
Step 11	**interface** *type number* **external** **Example:** `Router(config-oer-mc-br)# interface Ethernet 1/0 external`	Configures a border router interface as an OER-managed external interface. • External interfaces are used to forward traffic and for active monitoring. • A minimum of two external border router interfaces are required in an OER-managed network. At least one external interface must be configured on each border router. A maximum of 20 external interfaces can be controlled by single master controller. **Tip** Configuring an interface as an OER-managed external interface on a router enters OER border exit interface configuration mode. In this mode, you can configure maximum link utilization or cost-based optimization for the interface. **Note** Entering the **interface** command without the **external** or **internal** keyword places the router in global configuration mode and not OER border exit configuration mode. The **no** form of this command should be applied carefully so that active interfaces are not removed from the router configuration.

	Command or Action	Purpose

Step 12	**max-xmit-utilization** {**absolute** *kbps* | **percentage** *value*}
	Example:
	`Router(config-oer-mc-br-if)# max-xmit-utilization percentage 70`
		Configures the maximum utilization on a single OER managed exit link.
		• Use the **absolute** keyword and *kbps* argument to specify the absolute maximum utilization on an OER managed exit link in kbps.
		• Use the **percentage** keyword and *value* argument to specify percentage utilization of an exit link.

Step 13	**exit** **Example:** `Router(config-oer-mc-br-if)# exit`	Exits OER-managed border exit interface configuration mode and returns to OER-managed border router configuration mode.
Step 14	**exit** **Example:** `Router(config-oer-mc-br)# exit`	Exits OER-managed border router configuration mode and returns to OER master controller configuration mode.
Step 15	Repeat Step 10 through Step 14 with appropriate changes to set a utilization threshold for each external link.	—
Step 16	**keepalive** *timer* **Example:** `Router(config-oer-mc)# keepalive 10`	(Optional) Configures the length of time that an OER master controller will maintain connectivity with an OER border router after no keepalive packets have been received. • The example sets the keepalive timer to 10 seconds. The default keepalive timer is 60 seconds.

Step 17	**end**
	Example:
	`Router(config-oer-mc-learn)# end`
		Exits OER Top Talker and Top Delay learning configuration mode and returns to privileged EXEC mode.

Implementing Performance Routing Link Groups

Perform this task on a master controller to set up some performance routing link groups by identifying an exit link on a border router as a member of a link group, and to create an OER map to specify link groups for traffic classes defined in an OER policy. In this task, a link group is set up for video traffic and a set of high bandwidth exit links are identified as members of the video link group which is identified as a primary link group. A fallback link group is also specified.

An OER policy is created using an OER map where the primary and fall link groups are specified for traffic classes matching the OER map criteria. OER probes both the primary and fallback group links and selects the best link in the primary link group for the traffic class specified in this task. If none of the primary links are within policy, OER selects the bast link from the fallback group. For more details about link groups, see the "Performance Routing Link Grouping" section on page 9.

Prerequisites

This task requires the master controller and border routers to be running Cisco IOS Release 12.4(15)T, or later release.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. oer master
4. **border** *ip-address* [**key-chain** *key-chain-name*]
5. **interface** *type number* **external**
6. **link-group** *link-group-name* [*link-group-name* [*link-group-name*]]
7. **exit**
8. Repeat Step 5 through Step 7 with appropriate changes to set up link groups for all the external interface.
9. **interface** *type number* **internal**
10. **exit**
11. **ip access-list** {**standard** | **extended**} *access-list-name*
12. [*sequence-number*] **permit udp** *source source-wildcard* [*operator* [*port*]] *destination destination-wildcard* [*operator* [*port*]] [**dscp** *dscp-value*]
13. Repeat Step 12 for every required access list entry.
14. **exit**
15. **oer-map** *map-name sequence-number*
16. **match traffic-class access-list** *access-list-name*
17. **set link-group** *link-group-name* [**fallback** *link-group-name*]
18. **end**
19. **show oer master link-group** [*link-group-name*]

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`oer master`**
	Example:
	`Router(config)# oer master`
		Enters OER master controller configuration mode to configure a router as a master controller.
		• A master controller and border router process can be enabled on the same router (for example, in a network that has a single router with two exit links to different service providers).
		Note Only the syntax used in this context is displayed. For more details, see the *Cisco IOS Optimized Edge Routing Command Reference*.

Step 4	**`border`** *`ip-address`* `[`**`key-chain`** *`key-chain-name`*`]`
	Example:
	`Router(config-oer-mc)# border 192.168.1.2 key-chain border1_OER`
		Enters OER-managed border router configuration mode to establish communication with a border router.
		• An IP address is configured to identify the border router.
		• At least one border router must be specified to create an OER-managed network. A maximum of ten border routers can be controlled by a single master controller.
		• The value for the *key-chain-name* argument must match the key-chain name configured when the border router is set up.
		Note The **key-chain** keyword and *key-chain-name* argument must be entered when a border router is initially configured. However, this keyword is optional when reconfiguring an existing border router.

	Command or Action	Purpose

Step 5	`interface type number external`
	Example:
	`Router(config-oer-mc-br)# interface Serial 2/0 external`
		Configures a border router interface as an OER-managed external interface.
		• External interfaces are used to forward traffic and for active monitoring.
		• A minimum of two external border router interfaces are required in an OER-managed network. At least one external interface must be configured on each border router. A maximum of 20 external interfaces can be controlled by single master controller.
		Tip Configuring an interface as an OER-managed external interface on a router enters OER border exit interface configuration mode. In this mode, you can configure maximum link utilization or cost-based optimization for the interface.
		Note Entering the **interface** command without the **external** or **internal** keyword places the router in global configuration mode and not OER border exit configuration mode. The **no** form of this command should be applied carefully so that active interfaces are not removed from the router configuration.

Step 6	`link-group link-group-name [link-group-name [link-group-name]]`
	Example:
	`Router(config-oer-mc-br-if)# link-group VIDEO`
		Configures an OER border router exit interface as a member of a link group.
		• Use the *link-group-name* to specify the link group name for the interface.
		• Up to three link groups can be specified for each interface.
		• In this example, the Serial 2/0 external interface is configured as a member of the link group named VIDEO.
		Note The **link-group** command associates a link group with an interface. Another step, Step 17, uses the **set link-group** command to identify the link group as a primary or fallback group for traffic classes defined in an OER map.

Step 7	`exit`
	Example:
	`Router(config-oer-mc-br-if)# exit`
		Exits OER-managed border exit interface configuration mode and returns to OER-managed border router configuration mode.

Step 8	Repeat Step 5 through Step 7 with appropriate changes to set up link groups for all the external interface.	—

	Command or Action	Purpose

Step 9	**`interface`** *`type number`* **`internal`**
	Example:
	`Router(config-oer-mc-br)# interface FastEthernet 0/1 internal`
		Configures a border router interface as an OER controlled internal interface.
		• Internal interfaces are used for passive monitoring only. Internal interfaces do not forward traffic.
		• At least one internal interface must be configured on each border router.
		Note Support to configure a VLAN interface as an internal interface was introduced in Cisco IOS Release 12.3(14)T and 12.2(33)SRB.

Step 10	**`exit`**
	Example:
	`Router(config-oer-mc-br)# exit`
		Exits OER-managed border configuration mode and returns to global configuration mode.

Step 11	**`ip access-list`** `{`**`standard`** `|` **`extended`**`}` *`access-list-name`*
	Example:
	`Router(config)# ip access-list extended ACCESS_VIDEO`
		Defines an IP access list by name and enters extended named access list configuration mode.
		• OER supports only named access lists.
		• The example creates an extended IP access list named ACCESS_VIDEO.

Step 12	`[`*`sequence-number`*`]` **`permit udp`** *`source source-wildcard`* `[`*`operator`* `[`*`port`*`]]` *`destination destination-wildcard`* `[`*`operator`* `[`*`port`*`]]` `[`**`dscp`** *`dscp-value`*`]`
	Example:
	`Router(config-ext-nacl)# permit tcp any any 500`
		Sets conditions to allow a packet to pass a named IP access list.
		• The example is configured to identify all TCP traffic from any destination or source and from destination port number of 500. This specific TCP traffic is to be optimized.
		Note Only the syntax applicable to this task is shown. For more details, see the *Cisco IOS IP Application Services Command Reference*, Release 12.4T.

Step 13	Repeat Step 12 for more access list entries, as required.	—

Step 14	**`exit`**
	Example:
	`Router(config-ext-nacl)# exit`
		(Optional) Exits extended named access list configuration mode and returns to global configuration mode.

Step 15	**`oer-map`** *`map-name sequence-number`*
	Example:
	`Router(config)# oer-map VIDEO_MAP 10`
		Enters OER map configuration mode to configure an OER map.
		• *Only one match clause can be configured for each OER map sequence.*
		• Permit sequences are first defined in an IP prefix list and then applied with the **match ip address** (OER) command in Step 16.
		• The example creates an OER map named VIDEO_MAP.

	Command or Action	Purpose
Step 16	**match traffic-class access-list** *access-list-name* **Example:** `Router(config-oer-map)# traffic-class access-list ACCESS_VIDEO`	Manually configures an access list as match criteria used to create traffic classes using an OER map. • Each access list entry must contain a destination prefix and may include other optional parameters. • The example defines a traffic class using the criteria defined in the access list named ACCESS_VIDEO.
Step 17	**set link-group** *link-group-name* [**fallback** *link-group-name*] **Example:** `Router(config-oer-map)# set link-group video fallback voice`	Specifies a link group for traffic classes defined in an OER map to create an OER policy. • Use the *link-group-name* to specify the primary link group name for the policy. • Use the **fallback** keyword to specify the fallback link group name for the policy. • The example specifies the VIDEO link group as the primary link group for the traffic class matching the access list ACCESS_VIDEO. The link group VOICE is specified as the fallback link group.

Step 18	**end**
	Example:
	`Router(config-oer-map)# end`
		(Optional) Exits OER map configuration mode and returns to privileged EXEC mode.

Step 19	**show oer master link-group** [*link-group-name*] **Example:** `Router# show oer master link-group`	Displays information about configured OER link groups. • Use the optional *link-group-name* argument to display information for the specified OER link group. • If the *link-group-name* argument is not specified, information about all OER link groups is displayed. • The example displays information about all configured link groups.

Examples

The example output from the **show oer master link-group** command displays information about performance routing link groups configured using OER. In this example, the VIDEO link group is shown with other configured link groups.

```
Router# show oer master link-group

link group video
  Border           Interface        Exit id
  192.168.1.2      Serial2/0        1
 link group voice
  Border           Interface        Exit id
  192.168.1.2      Serial2/0        1
  192.168.1.2      Serial3/0        2
  192.168.3.2      Serial4/0        4
 link group data
  Border           Interface        Exit id
  192.168.3.2      Serial3/0        3
```

Configuring OER Cost-Based Policies

Perform this task to configure cost-based optimization. Cost-based optimization is configured on a master controller using the **cost-minimization** command in OER border exit interface configuration mode (under the external interface configuration). Cost-based optimization supports tiered and fixed billing methods.

Prerequisites

This task requires the master controller and border routers to be running Cisco IOS Release 12.3(14)T, 12.2(33)SRB, or later releases.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **oer master**
4. **border** *ip-address* [**key-chain** *key-chain-name*]
5. **interface** *type number* **external**
6. **cost-minimization** {**calc** {**combined** | **separate** | **sum**} | **discard** [**daily**] {**absolute** *number* | **percent** *percentage*} | **end day-of-month** *day* [**offset** *hh*:*mm*] | **fixed fee** [*cost*] | **nickname** *name* | **sampling period** *minutes* [**rollup** *minutes*] | **summer-time** *start end* [*offset*] | **tier** *percentage fee*}
7. Repeat Step 6 to configure additional cost-based optimization policies, if required.
8. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`oer master`**
	Example:
	`Router(config)# oer master`
		Enters OER master controller configuration mode to configure global prefix and exit link policies.

Step 4	**`border`** *`ip-address`* `[`**`key-chain`** *`key-chain-name`*`]`
	Example:
	`Router(config-oer-mc)# border 10.100.1.1 key-chain OER`
		Enters OER-managed border router configuration mode to establish communication with a border router.
		Note The **key-chain** keyword is required only for initial border router configuration.

	Command or Action	Purpose
Step 5	**interface** *type number* **external** **Example:** `Router(config-oer-mc-br)# interface ethernet 0/0 external`	Enters OER border exit interface configuration mode to configure a border router interface as an external interface. • At least one external interface must be configured on each border router.

Step 6	**cost-minimization** {**calc** {**combined** | **separate** | **sum**} | **discard** [**daily**] {**absolute** *number* | **percent** *percentage*} | **end day-of-month** *day* [**offset** *hh:mm*] | **fixed fee** [*cost*] | **nickname** *name* | **sampling period** *minutes* [**rollup** *minutes*] | **summer-time** *start end* [*offset*] | **tier** *percentage fee*}
	Example:
	`Router(config-oer-mc-br-if)# cost-minimization end day-of-month 30 offset 3:00`
		Configures cost-based optimization policies on a master controller.
		• Cost-based optimization supports fixed or tier-based billing, inbound and outbound cost measurements, and very granular sampling.
		• The **calc** keyword is used to configure how the fee is calculated. You can configure the master controller to combine ingress and egress samples, either by first adding and then combining or by analyzing ingress and egress samples separately.
		• The **discard** keyword is used to configure the number of samples that are removed for bursty link usage. It is specified as a percentage or as an absolute value. If a sampling rollup is configured, the discard values also applies to the rollup. If the **daily** keyword is entered, samples are analyzed and discarded on a daily basis. At the end of the billing cycle, monthly sustained usage is calculated by averaging daily sustained utilization.
		• The **end** keyword is used to configure the last day of the billing cycle. Entering the offset keyword allows you to adjust the end of the cycle to compensate for a service provider in a different zone.
		• The **fixed** keyword is configured when the service provider bills for network access over the specified exit link at a flat rate. The **fee** keyword is optionally used to specify the exit link cost.
		• The **nickname** keyword is used to apply label that identifies the service provider.
		• The **sampling** keyword is used to configure the time intervals at which link utilization samples are gathered. By default, the link is sampled every five minutes.
		• The **rollup** keyword is used to reduce the number of samples by aggregating them. All samples collected during the rollup period are averaged to calculate rollup utilization.
		Note The minimum number that can be entered for the rollup period must be equal to or greater than the number that is entered for the sampling period.
		• In this example, the billing end date is set to 30, and a a three-hour offset is applied.

	Command or Action	Purpose
Step 7	Repeat Step 6 to configure additional cost-based optimization policies, if required.	—

Step 8	`end`
	Example:
	`Router(config-oer-mc-br-if)# end`
		Exits OER border exit interface configuration mode and enters privileged EXEC mode.

Configuring OER Network Security Policies

Perform one of the following two optional tasks to help prevent and mitigate attacks on your network. The first task uses the black hole routing technique, and the second task uses the sinkhole routing technique.

- Configuring Black Hole Routing Using an OER Map, page 50
- Configuring Sinkhole Routing Using an OER Map, page 52

Configuring Black Hole Routing Using an OER Map

Perform this task to configure an OER map to filter packets to be forwarded to a null interface, meaning that the packets are discarded in a “black hole.” The prefix list is configured after an IP prefix is identified as the source of the attack on the network. Some protocols such as BGP allow the redistribution of back hole routes, but other protocols do not.

Prerequisites

This task requires the master controller and border routers to be running Cisco IOS Release 12.4(9)T, 12.2(33)SRB, or later releases.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network/length* | **permit** *network/length*} [**le** *le-value*]
4. **oer-map** *map-name sequence-number*
5. **match ip address** {**access-list** *access-list-name* | **prefix-list** *prefix-list-name*}
6. **set interface null0**
7. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network***/***length* | **permit** *network***/***length*} [**le** *le-value*] **Example:** `Router(config)# ip prefix-list BLACK_HOLE_LIST seq 10 permit 10.20.21.0/24`	Creates an IP prefix list. • IP prefix lists are used to manually select prefixes for monitoring by the OER master controller. • A master controller can monitor and control an exact prefix of any length including the default route. If an exact prefix is specified, OER monitors only the exact prefix. • A master controller can monitor and control an inclusive prefix using the **le** keyword set to 32. OER monitors the configured prefix and any more specific prefixes (for example, configuring the 10.0.0.0/8 le 32 prefix would include the 10.1.0.0/16 and the 10.1.1.0/24 prefixes) over the same exit and records the information in the routing information base (RIB). • The prefixes specified in the IP prefix list are imported into an OER map using the **match ip address** (OER) command. • The example creates an IP prefix list named BLACK_HOLE_LIST that permits prefixes from the 10.20.21.0/24 subnet.
Step 4	**oer-map** *map-name sequence-number* **Example:** `Router(config)# oer-map BLACK_HOLE_MAP 10`	Enters OER map configuration mode to configure an OER map to apply policies to selected IP prefixes. • *Only one match clause can be configured for each OER map sequence.* • Deny sequences are first defined in an IP prefix list and then applied with the **match ip address** (OER) command in the previous step. • The example creates an OER map named BLACK_HOLE_MAP.

	Command or Action	Purpose

Step 5	**match ip address** {**access-list** *access-list-name* | **prefix-list** *prefix-list-name*}
	Example:
	`Router(config-oer-map)# match ip address prefix-list BLACK_HOLE_LIST`
		References an extended IP access list or IP prefix as match criteria in an OER map.
		• Only a single match clause can be configured for each OER map sequence.
		• The example configures the IP prefix list named BLACK_HOLE_LIST as match criteria in an OER map.

Step 6	**set interface null0** **Example:** `Router(config-oer-map)# set interface null0`	Creates a set clause entry to forward packets to the null interface, meaning that they are discarded. • The example creates a set clause entry to specify that the packets matching the prefix list, BLACK_HOLE_LIST, are discarded.

Step 7	**end**
	Example:
	`Router(config-oer-map)# end`
		(Optional) Exits OER map configuration mode and returns to privileged EXEC mode.

Configuring Sinkhole Routing Using an OER Map

Perform this task to configure an OER map to filter packets to be forwarded to a next hop. The next hop is a router where the packets can be stored, analyzed, or discarded (the sinkhole analogy). The prefix list is configured after an IP prefix is identified as the source of an attack on the network.

Prerequisites

This task requires the master controller and border routers to be running Cisco IOS Release 12.4(9)T, 12.2(33)SRB, or later releases.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network/length* | **permit** *network/length*} [**le** *le-value*]
4. **oer-map** *map-name sequence-number*
5. **match ip address** {**access-list** *access-list-name* | **prefix-list** *prefix-list-name*}
6. **set next-hop** *ip-address*
7. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		- Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network***/***length* | **permit** *network***/***length*} [**le** *le-value*] **Example:** `Router(config)# ip prefix-list SINKHOLE_LIST seq 10 permit 10.20.21.0/24`	Creates an IP prefix list. - IP prefix lists are used to manually select prefixes for monitoring by the OER master controller. - A master controller can monitor and control an exact prefix of any length including the default route. If an exact prefix is specified, OER monitors only the exact prefix. - A master controller can monitor and control an inclusive prefix using the **le** keyword set to 32. OER monitors the configured prefix and any more specific prefixes (for example, configuring the 10.0.0.0/8 le 32 prefix would include the 10.1.0.0/16 and the 10.1.1.0/24 prefixes) over the same exit and records the information in the routing information base (RIB). - The prefixes specified in the IP prefix list are imported into an OER map using the **match ip address** (OER) command. - The example creates an IP prefix list named SINKHOLE_LIST that permits prefixes from the 10.20.21.0/24 subnet.
Step 4	**oer-map** *map-name sequence-number* **Example:** `Router(config-oer-mc)# oer-map SINKHOLE_MAP 10`	Enters OER map configuration mode to configure an OER map to apply policies to selected IP prefixes. - *Only one match clause can be configured for each OER map sequence.* - Deny sequences are first defined in an IP prefix list and then applied with the **match ip address** (OER) command in the previous step. - The example creates an OER map named SINKHOLE_MAP.

Step 5	**match ip address** {**access-list** *access-list-name* | **prefix-list** *prefix-list-name*}
	Example:
	`Router(config-oer-map)# match ip address prefix-list SINKHOLE_LIST`
		References an extended IP access list or IP prefix as match criteria in an OER map.
		- Only a single match clause can be configured for each OER map sequence.
		- The example configures the IP prefix list named SINKHOLE_LIST as match criteria in an OER map.

	Command or Action	Purpose

Step 6	**`set next-hop`** *`ip-address`*
	Example:
	`Router(config-oer-map)# set next-hop 10.20.21.6`
		Creates a set clause entry specifying that packets are forwarded to the next hop.
		• The example creates a set clause entry to specify that the packets matching the prefix list, SINKHOLE_LIST, are forwarded to the next hop at 10.20.21.6.

Step 7	**`end`**
	Example:
	`Router(config)# end`
		(Optional) Exits OER map configuration mode and returns to privileged EXEC mode.

Configuring OER Voice Traffic Optimization Using Active Probes

Support for optimizing voice traffic using OER was introduced in 12.4(6)T. Configuring OER to optimize voice traffic using active probes involves several decisions and subsequent branching tasks. The first step is to identify the traffic to be optimized and decide whether to use a prefix list or an access list. Use a prefix list to identify all traffic, including voice traffic, with a specific set of destination prefixes. Use an access list to identify only voice traffic with a specific destination prefix and carried over a specific protocol.

The second step in optimizing voice traffic is to configure active probing using the **active-probe** or **set active-probe** command to specify the type of active probe to be used. In Cisco IOS Release 12.4(6)T, 12.2(33)SRB, the ability to set a forced target assignment for the active probe was introduced.

The final step in optimizing voice traffic is to configure an OER policy to set the performance metrics that you want OER to apply to the identified traffic.

Perform one of the first two optional tasks, depending on whether you want to use a prefix list or an access list to identify the traffic to be optimized. The third task can be used with traffic identified using an access list, and it also demonstrates how to use a forced target assignment. For an example configuration that can be used with traffic identified using a prefix list, see the "Optimizing Only Voice Traffic Using Active Probes" section on page 69.

- Identifying Traffic for OER Using a Prefix List, page 54
- Identifying Voice Traffic to Optimize Using an Access List, page 55
- Configuring OER Voice Probes with a Target Assignment, page 57

Identifying Traffic for OER Using a Prefix List

Before traffic can be measured using OER, it must be identified. Perform this task to use a prefix list to identify the traffic that OER will probe.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network/length* | **permit** *network/length*} [**le** *le-value*]
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network***/***length* | **permit** *network***/***length*} [**le** *le-value*] **Example:** `Router(config)# ip prefix-list TRAFFIC_PFX_LIST seq 10 permit 10.20.21.0/24`	Creates an IP prefix list. • IP prefix lists are used to manually select prefixes for monitoring by the OER master controller. • A master controller can monitor and control an exact prefix of any length including the default route. If an exact prefix is specified, OER monitors only the exact prefix. • A master controller can monitor and control an inclusive prefix using the **le** keyword set to 32. OER monitors the configured prefix and any more specific prefixes (for example, configuring the 10.0.0.0/8 le 32 prefix would include the 10.1.0.0/16 and the 10.1.1.0/24 prefixes) over the same exit and records the information in the routing information base (RIB). • The prefixes specified in the IP prefix list are imported into an OER map using the **match ip address** (OER) command. • The example creates an IP prefix list named TRAFFIC_PFX_LIST that permits prefixes from the 10.20.21.0/24 subnet.
Step 4	**exit** **Example:** `Router(config)# exit`	(Optional) Exits global configuration mode and returns to privileged EXEC mode.

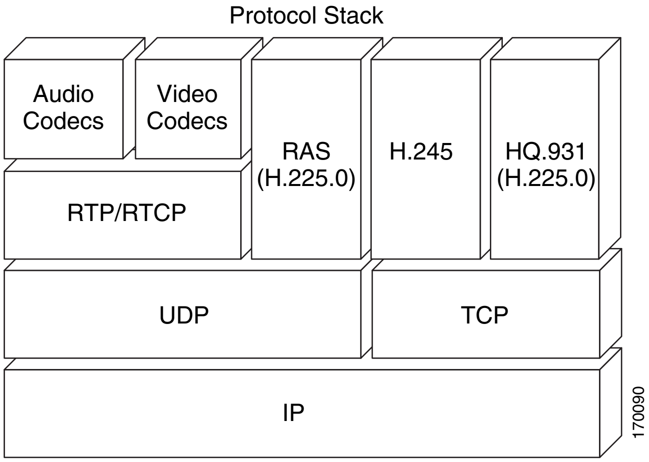
Identifying Voice Traffic to Optimize Using an Access List

Perform this task to use an access list to identify the voice traffic. Before voice traffic can be optimized, it must be identified. Voice traffic that has to be optimized must be configured because OER does not “learn” about voice traffic on IP networks during an OER learn phase.

IP Protocol Stack for Voice

Voice traffic uses a variety of protocols and streams on the underlying IP network. Figure 5 is a representation of the protocol options available for carrying voice traffic over IP. Most signaling traffic for voice is carried over TCP. Most voice calls are carried over User Datagram Protocol (UDP) and Real-Time Protocol (RTP). You can configure your voice devices to use a specific range of destination port numbers over UDP to carry voice call traffic.

Figure 5 Protocol Stack Options Available for Voice Traffic



Prerequisites

This task requires the master controller and border routers to be running Cisco IOS Release 12.4(6)T, 12.2(33)SRB, or later release.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access list** {**standard** | **extended**} *access-list-name*
4. [*sequence-number*] **permit udp** *source source-wildcard* [*operator* [*port*]] *destination destination-wildcard* [*operator* [*port*]]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip access-list** {**standard** | **extended**} *access-list-name* **Example:** `Router(config)# ip access-list extended VOICE_ACCESS_LIST`	Defines an IP access list by name. • OER supports only named access lists. • The example creates an extended IP access list named VOICE_ACCESS_LIST.
Step 4	[*sequence-number*] **permit udp** *source source-wildcard* [*operator* [*port*]] *destination destination-wildcard* [*operator* [*port*]] **Example:** `Router(config-ext-nacl)# permit udp any range 16384 32767 10.20.20.0 0.0.0.15 range 16384 32767`	Sets conditions to allow a packet to pass a named IP access list. • The example is configured to identify all UDP traffic ranging from a destination port number of 16384 to 32767 from any source to a destination prefix of 10.20.20.0/24. This specific UDP traffic is to be optimized. • Only the syntax applicable to this task is shown. For more details, see the *Cisco IOS IP Application Services Command Reference*.

Step 5	**end**
	Example:
	`Router(config-ext-nacl)# end`
		(Optional) Exits extended access list configuration mode and returns to privileged EXEC mode.

Configuring OER Voice Probes with a Target Assignment

After identifying the traffic (in this example, voice traffic identified using an access list) to be optimized, perform this task to configure the OER jitter probes and assign the results of the jitter probes to optimize the identified traffic. In this task, the OER active voice probes are assigned a forced target for OER instead of the usual longest match assigned target. Before configuring the OER jitter probe on the source device, the IP SLAs Responder must be enabled on the target device (the operational target). The IP SLAs Responder is available only on Cisco IOS software-based devices. Start this task at the network device that runs the IP SLAs Responder.

Note The device that runs the IP SLAs Responder does not have to be configured for OER.

Note Policies applied in an OER map do not override global policy configurations.

Prerequisites

- Before configuring this task, perform the "Identifying Voice Traffic to Optimize Using an Access List" section on page 55.
- This task requires the master controller and border routers to be running Cisco IOS Release 12.4(6)T, 12.2(33)SRB, or later releases.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla monitor responder**
4. **exit**
5. Move to the network device that is the OER master controller.
6. **enable**
7. **configure terminal**
8. **oer-map** *map-name sequence-number*
9. **match ip address** {**access-list** *access-list-name* | **prefix-list** *prefix-list-name*}
10. **set active probe** *probe-type ip-address* [**target-port** *number*] [**codec** *codec-name*]
11. **set probe frequency** *seconds*
12. **set jitter threshold** *maximum*
13. **set mos threshold** *minimum* **percent** *percent*
14. **set resolve** {**cost priority** *value* | **delay priority** *value* **variance** *percentage* | **jitter priority** *value* **variance** *percentage* | **loss priority** *value* **variance** *percentage* | **mos priority** *value* **variance** *percentage* | **range priority** *value* | **utilization priority** *value* **variance** *percentage*}
15. **set resolve mos priority** *value* **variance** *percentage*
16. **set delay** {**relative** *percentage* | **threshold** *maximum*}
17. **exit**
18. **oer master**
19. **policy-rules** *map-name*
20. **end**
21. **show oer master active-probes forced**
22. **show oer master policy** [*sequence-number* | *policy-name* | **default]**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** Router> enable	Enables privileged EXEC mode. - Enter your password if prompted.
Step 2	**configure terminal** **Example:** Router# configure terminal	Enters global configuration mode.
Step 3	**ip sla monitor responder** **Example:** Router(config)# ip sla monitor responder	Enables the IP SLAs Responder.
Step 4	**exit** **Example:** Router(config)# exit	Exits global configuration mode and returns to privileged EXEC mode.
Step 5	Move to the network device that is the OER master controller.	—
Step 6	**enable** **Example:** Router> enable	Enables privileged EXEC mode. - Enter your password if prompted.
Step 7	**configure terminal** **Example:** Router# configure terminal	Enters global configuration mode.
Step 8	**oer-map** *map-name sequence-number* **Example:** Router(config)# oer-map TARGET_MAP 10	Enters OER map configuration mode to configure an OER map to apply policies to selected IP prefixes. - *Only one match clause can be configured for each OER map sequence.* - Deny sequences are first defined in an IP prefix list and then applied with the **match ip address** (OER) command in Step 9. - The example creates an OER map named TARGET_MAP.
Step 9	**match ip address** {**access-list** *access-list-name* | **prefix-list** *prefix-list-name*} **Example:** Router(config-oer-map)# match ip address access-list VOICE_ACCESS_LIST	References an extended IP access list or IP prefix as match criteria in an OER map. - Only a single match clause can be configured for each OER map sequence. - The example configures the IP access list named VOICE_ACCESS_LIST as match criteria in an OER map. The access list was created in the "Identifying Voice Traffic to Optimize Using an Access List" task.

	Command or Action	Purpose
Step 10	**set active-probe** *probe-type ip-address* [**target-port** *number*] [**codec** *codec-name*] **Example:** `Router(config-oer-map)# set active-probe jitter 10.20.22.1 target-port 2000 codec g729a`	Creates a set clause entry to assign a target prefix for an active probe. • Use the *probe-type* argument to specify one four probe types: echo, jitter, tcp-conn, or udp-echo. • The *ip-address* argument to specify the target IP address of a prefix to be monitored using the specified type of probe. • The **target-port** keyword and *number* argument are used to specify the destination port number for the active probe. • The **codec** keyword and *codec-name* argument are used only with the jitter probe type. Specifies the codec value used for Mean Opinion Score (MOS) calculation. The codec values must be one of the following: g711alaw, g711ulaw, or g729a. • The example creates a set clause entry to specify the target IP address of a prefix and a specific port number to actively monitor using jitter.
Step 11	**set probe frequency** *seconds* **Example:** `Router(config-oer-map)# set probe frequency 10`	Creates a set clause entry to set the frequency of the OER active probe. • The *seconds* argument is used to set the time, in seconds, between the active probe monitoring of the specified IP prefixes. • The example creates a set clause to set the active probe frequency to 10 seconds.
Step 12	**set jitter threshold** *maximum* **Example:** `Router(config-oer-map)# set jitter threshold 20`	Creates a set clause entry to configure the jitter threshold value. • The **threshold** keyword is used to configure the maximum jitter value, in milliseconds. • The example creates a set clause that sets the jitter threshold value to 20 for traffic that is matched in the same OER map sequence.

	Command or Action	Purpose

Step 13	**set mos** {**threshold** *minimum* **percent** *percent*}
	Example:
	`Router(config-oer-map)# set mos threshold 4.0 percent 30`
		Creates a set clause entry to configure the MOS threshold and percentage values used to decide whether an alternate exit is be selected.
		• The **threshold** keyword is used to configure the minimum MOS value.
		• The **percent** keyword is used to configure the percentage of MOS values that are below the MOS threshold.
		• OER calculates the percentage of MOS values below the MOS threshold that are recorded in a five-minute period. If the percentage value exceeds the configured percent value or the default value, the master controller searches for alternate exit links.
		• The example creates a set clause that sets the threshold MOS value to 4.0 and the percent value to 30 percent for traffic that is matched in the same OER map sequence.

Step 14	**set resolve** {**cost priority** *value* | **delay priority** *value* **variance** *percentage* | **jitter priority** *value* **variance** *percentage* | **loss priority** *value* **variance** *percentage* | **mos priority** *value* **variance** *percentage* | **range priority** *value* | **utilization priority** *value* **variance** *percentage*}
	Example:
	`Router(config-oer-map)# set resolve jitter priority 1 variance 10`
		Creates a set clause entry to configure policy priority or resolve policy conflicts.
		• This command is used to set priority for a policy type when multiple policies are configured for the same prefix. When this command is configured, the policy with the highest priority will be selected to determine the policy decision.
		• The **priority** keyword is used to specify the priority value. Configuring the number 1 assigns the highest priority to a policy. Configuring the number 10 assigns the lowest priority.
		• Each policy must be assigned a different priority number.
		• The **variance** keyword is used to set an allowable variance for a user-defined policy. This keyword configures the allowable percentage that an exit link or prefix can vary from the user-defined policy value and still be considered equivalent.
		• Variance cannot be configured for cost or range policies.
		• The example creates set clause that configures the priority for jitter policies to 1 for voice traffic. The variance is configured to allow a 10 percent difference in jitter statistics before a prefix is determined to be out-of-policy.

	Command or Action	Purpose
Step 15	**set resolve mos priority** *value* **variance** *percentage* **Example:** `Router(config-oer-map)# set resolve mos priority 2 variance 15`	Creates a set clause entry to configure policy priority or resolve policy conflicts. • The example creates set clause that configures the priority for MOS policies to 2 for voice traffic. The variance is configured to allow a 15 percent difference in MOS values before a prefix is determined to be out-of-policy. **Note** Only the syntax applicable to this task is used in this example. For more details, see Step 14.

Step 16	**set delay** {**relative** *percentage* | **threshold** *maximum*}
	Example:
	`Router(config-oer-map)# set delay threshold 100`
		Creates a set clause entry to configure the delay threshold.
		• The delay threshold can be configured as a relative percentage or as an absolute value for match criteria.
		• The **relative** keyword is used to configure a relative delay percentage. The relative delay percentage is based on a comparison of short-term and long-term measurements.
		• The **threshold** keyword is used to configure the absolute maximum delay period in milliseconds.
		• The example creates a set clause that sets the absolute maximum delay threshold to 100 milliseconds for traffic that is matched in the same OER map sequence.

Step 17	**exit** **Example:** `Router(config-oer-map)# exit`	Exits OER map configuration mode and returns to global configuration mode.
Step 18	**oer master** **Example:** `Router(config)# oer master`	Enters OER master controller configuration mode to configure a router as a master controller. • A master controller and border router process can be enabled on the same router (for example, in a network that has a single router with two exit links to different service providers). **Note** Only the syntax used in this context is displayed. For more details, see the *Cisco IOS Optimized Edge Routing Command Reference*.
Step 19	**policy-rules** *map-name* **Example:** `Router(config-oer-mc)# policy-rules TARGET_MAP`	Applies a configuration from an OER map to a master controller configuration in OER master controller configuration mode. • Reentering this command with a new OER map name will immediately overwrite the previous configuration. This behavior is designed to allow you to quickly select and switch between predefined OER maps. • The example applies the configuration from the OER map named TARGET_MAP.

	Command or Action	Purpose

Step 20	**end**
	Example:
	`Router(config-oer-mc)# end`
		Exits OER master controller configuration mode and enters privileged EXEC mode.

Step 21	**show oer master active-probes** [**appl** | **forced**] **Example:** `Router# show oer master active-probes forced`	Displays connection and status information about active probes on an OER master controller. • The output from this command displays the active probe type and destination, the border router that is the source of the active probe, the target prefixes that are used for active probing, and whether the probe was learned or configured. • The **appl** keyword is used to filter the output to display information about applications optimized by the master controller. • The **forced** keyword is used to show any forced targets that are assigned. • The example displays connection and status information about the active probes generated for voice traffic configured with a forced target assignment.
Step 22	**show oer master policy** [*sequence-number* | *policy-name* | **default**] **Example:** `Router# show oer master policy TARGET_MAP`	Displays policy settings on an OER master controller. • The output of this command displays default policy and policies configured with an OER map. • The *sequence-number* argument is used to display policy settings for the specified OER map sequence. • The *policy-name* argument is used to display policy settings for the specified OER policy map name. • The **default** keyword is used to display only the default policy settings. • The example displays the policy settings configured for the TARGET_MAP policy.

Examples

This example shows output from the **show oer master active-probes forced** command. The output is filtered to display only connection and status information about the active probes generated for voice traffic configured with a forced target assignment.

```
Router# show oer master active-probes forced

OER Master Controller active-probes
Border   = Border Router running this Probe
Policy   = Forced target is configure under this policy
Type     = Probe Type
Target   = Target Address
TPort    = Target Port
N - Not applicable

The following Forced Probes are running:
```

```
Border          State    Policy              Type     Target          TPort
10.20.20.2      ACTIVE   40                  jitter   10.20.22.1      3050
10.20.21.3      ACTIVE   40                  jitter   10.20.22.4      3050
```

What to do Next

For further configuration examples of OER voice traffic optimization, see the "Configuring OER Voice Traffic Optimization Using Active Probes: Examples" section on page 69.

Configuration Examples for Configuring and Applying OER Policies

The following examples in this section show various OER policy configurations:

- Configuring and Applying an OER Policy to Learned Traffic Classes: Example, page 64
- Configuring and Applying an OER Policy to Configured Traffic Classes: Example, page 65
- Preventing OER Optimization of Learned Prefixes: Example, page 65
- Configuring and Applying an OER Policy to Learned Inside Prefixes: Example, page 65
- Configuring and Applying an OER Policy to Configured Inside Prefixes: Example, page 66
- Configuring Policy Rules for OER Maps: Example, page 66
- Configuring Multiple OER Policy Conflict Resolution: Example, page 66
- Configuring an Exit Link Load Balancing OER Policy: Example, page 67
- Implementing Performance Routing Link Groups: Example, page 67
- Configuring OER Cost-Based Policies: Example, page 68
- Configuring OER Network Security Policies: Examples, page 68
- Configuring OER Voice Traffic Optimization Using Active Probes: Examples, page 69

Configuring and Applying an OER Policy to Learned Traffic Classes: Example

The following example uses learned traffic classes and overwrites many of the default policy settings and configures the master controller to move traffic classes to the best available exit link when any of the configured or default policy settings exceed their thresholds:

```
enable
configure terminal
oer master
 backoff 200 2000 200
 delay threshold 2000
 holddown 400
 loss threshold 1500
 periodic 180
 unreachable threshold 1000
 mode select-exit best
 end
```

Configuring and Applying an OER Policy to Configured Traffic Classes: Example

The following example uses traffic classes filtered by a prefix list and an access list and overwrites some of the default policy settings. The policies are configured using two OER maps that apply to different traffic classes that represent voice traffic. The master controller is configured to move traffic classes to the first in-policy exit link when any of the configured or default policy settings exceed their thresholds. To run this task, both the master controller and border routers must be running Cisco IOS Release 12.4(9)T, 12.2(33)SRB, or later releases.

```
enable
configure terminal
ip prefix-list CONFIG_TRAFFIC_CLASS seq 10 permit 10.1.5.0/24
ip access-list extended VOICE_TRAFFIC_CLASS
 permit udp any range 16384 32767 10.1.5.0 0.0.0.15 range 16384 32767 dscp ef
 exit
oer-map CONFIG_MAP 10
 match ip address prefix-list CONFIG_TRAFFIC_CLASS
 set backoff 100 1000 100
 set delay threshold 1000
 set loss relative 25
 set periodic 360
 set unreachable relative 20
 exit
oer-map VOICE_MAP 10
 match ip address access-list VOICE_TRAFFIC_CLASS
 set active-probe jitter 10.1.5.1 target-port 2000 codec g729a
 set probe-frequency 20
 set jitter threshold 30
 set mos threshold 4.0 percent 25
 set mode select-exit good
 end
```

Preventing OER Optimization of Learned Prefixes: Example

The following example shows how to configure OER to prevent specified prefixes being optimized. In this example, an IP prefix list is created with two entries for different prefixes that are not to be optimized. An OER map is configured with two entries in a sequence that will prevent OER from optimizing the prefixes specified in the prefix list, although the prefixes may be learned. If the sequence numbers of the OER map entries are reversed, OER will learn and attempt to optimize the prefixes.

```
enable
configure terminal
ip prefix-list DENY_PREFIX deny 172.17.10.0/24
ip prefix-list DENY_PREFIX deny 172.19.10.0/24
oer-map DENY_PREFIX_MAP 10
 match ip address prefix-list DENY_PREFIX
 exit
oer-map DENY_PREFIX_MAP 20
 match oer learn throughput
 end
```

Configuring and Applying an OER Policy to Learned Inside Prefixes: Example

The following example shows how to apply an OER policy to learned inside prefixes:

```
enable
configure terminal
oer-map INSIDE_LEARN 10
```

```
 match oer learn inside
 set delay threshold 2000
 set loss relative 20
 set unreachable relative 90
 end
```

Configuring and Applying an OER Policy to Configured Inside Prefixes: Example

The following example shows how to create an OER map named INSIDE_CONFIGURE and apply an OER policy to manually configured inside prefixes:

```
enable
configure terminal
 oer-map INSIDE_CONFIGURE 10
 match ip address prefix-list INSIDE_PREFIXES inside
 set delay threshold 2000
 set loss relative 20
 set unreachable relative 80
 end
```

Configuring Policy Rules for OER Maps: Example

The following example shows how to configure the **policy-rules** command to apply the OER map configuration named BLUE under OER master controller mode:

```
enable
configure terminal
oer-map BLUE 10
 match oer learn delay
 set loss relative 90
 exit
oer master
 policy-rules BLUE
 exit
```

Configuring Multiple OER Policy Conflict Resolution: Example

The following example configures an OER resolve policy that sets delay to the highest priority, followed by loss, and then utilization. The delay policy is configured to allow a 20 percent variance, the loss policy is configured to allow a 30 percent variance, and the utilization policy is configured to allow a 10 percent variance.

```
enable
configure terminal
oer master
 resolve delay priority 1 variance 20
 resolve loss priority 2 variance 30
 resolve utilization priority 3 variance 10
 end
```

Configuring an Exit Link Load Balancing OER Policy: Example

The following example configures an OER load balancing policy for traffic class flows over the border router exit links. This example task is performed at the master controller and configures an exit link utilization range and an exit link utilization threshold with policy priorities set for utilization and range policies. Performance policies, delay and loss, are disabled. OER uses both the utilization and range thresholds to load balance the traffic flow over the exit links.

```
enable
configure terminal
oer master
 max-range-utilization percentage 25
 mode select-exit best
 resolve range priority 1
 resolve utilization priority 2 variance 15
 no resolve delay
 no resolve loss
 border 10.1.4.1
 interface Ethernet 1/0 external
 max-xmit-utilization absolute 10000
 exit
 exit
 border 10.1.2.1
 interface Ethernet 1/0 external
 max-xmit-utilization absolute 10000
 end
```

Implementing Performance Routing Link Groups: Example

The following example shows how to implement link groups. In this example, an OER map named VIDEO_MAP is created to configure OER to define a traffic class that matches an access list named ACCESS_VIDEO. The traffic class is configured to use a link group named VIDEO as the primary link group, and a fallback group named VOICE. The VIDEO link group may be a set of high bandwidth links that are preferred for video traffic.

```
enable
configure terminal
border 10.1.4.1
 interface serial 2/0 external
  link-group VIDEO
  exit
 interface serial 3/0 external
  link-group VOICE
  exit
 interface Ethernet 1/0 internal
 exit
ip access-list extended ACCESS_VIDEO
 permit tcp any 10.1.1.0 0.0.0.255 eq 500
 permit tcp any 172.17.1.0 0.0.255.255 eq 500
 permit tcp any 172.17.1.0 0.0.255.255 range 700 750
 permit tcp 192.168.1.1 0.0.0.0 10.1.2.0 0.0.0.255 eq 800 any any dscp ef
 exit
oer-map VIDEO_MAP 10
 match traffic-class access-list ACCESS_VIDEO
 set link-group VIDEO fallback VOICE
 end
```

Configuring OER Cost-Based Policies: Example

The following example shows how to configure cost-based optimization on a master controller. Cost optimization configuration is applied under the external interface configuration. In this example, a policy for a tiered billing cycle is configured that sets a tiered fee of 1000 at 100 percent utilization, a tiered fee of 900 at 90 percent utilization, and a tiered fee of 800 at 80 percent utilization. Calculation is configured separately for egress and ingress samples. The time interval between sampling is set to 10 minutes and these samples are configured to be rolled up every 60 minutes.

```
enable
configure terminal
oer master
 border 10.5.5.55 key-chain key
 interface Ethernet 0/0 external
 cost-minimization nickname ISP1
 cost-minimization end day-of-month 30 180
 cost-minimization calc separate
 cost-minimization sampling 10 rollup 60
 cost-minimization tier 100 fee 1000
 cost-minimization tier 90 fee 900
 cost-minimization tier 80 fee 800
 exit
```

Configuring OER Network Security Policies: Examples

Black Hole Routing Example

The following example creates an OER map named BLACK_HOLE_MAP that matches traffic defined in the IP prefix list named PREFIX_BLACK_HOLE. The OER map filters packets to be forwarded to a null interface, meaning that the packets are discarded in a “black hole.” The prefix list is configured after an IP prefix is identified as the source of the attack on the network.

```
enable
configure terminal
ip prefix-list PREFIX_BLACK_HOLE seq 10 permit 10.1.5.0/24
oer-map BLACK_HOLE_MAP 10
 match ip address prefix-list PREFIX_BLACK_HOLE
 set interface null0
 end
```

Sink Hole Routing Example

The following example creates an OER map named SINK_HOLE_MAP that matches traffic defined in the IP prefix list named PREFIX_SINK_HOLE. The OER map filters packets to be forwarded to a next hop. The next hop is a router where the packets can be stored, analyzed, or discarded (the sinkhole analogy). The prefix list is configured after an IP prefix is identified as the source of an attack on the network.

```
enable
configure terminal
ip prefix-list PREFIX_SINK_HOLE seq 10 permit 10.1.5.0/24
oer-map SINK_HOLE_MAP 10
 match ip address prefix-list PREFIX_SINK_HOLE
 set next-hop 10.1.1.3
 end
```

Configuring OER Voice Traffic Optimization Using Active Probes: Examples

Voice packets traveling through an IP network are no different from data packets. In the plain old telephone system (POTS), voice traffic travels over circuit-switched networks with predetermined paths and each phone call is given a dedicated connection for the duration of the call. Voice traffic using POTS has no resource contention issues, but voice traffic over an IP network has to contend with factors such as delay, jitter, and packet loss, which can affect the quality of the phone call.

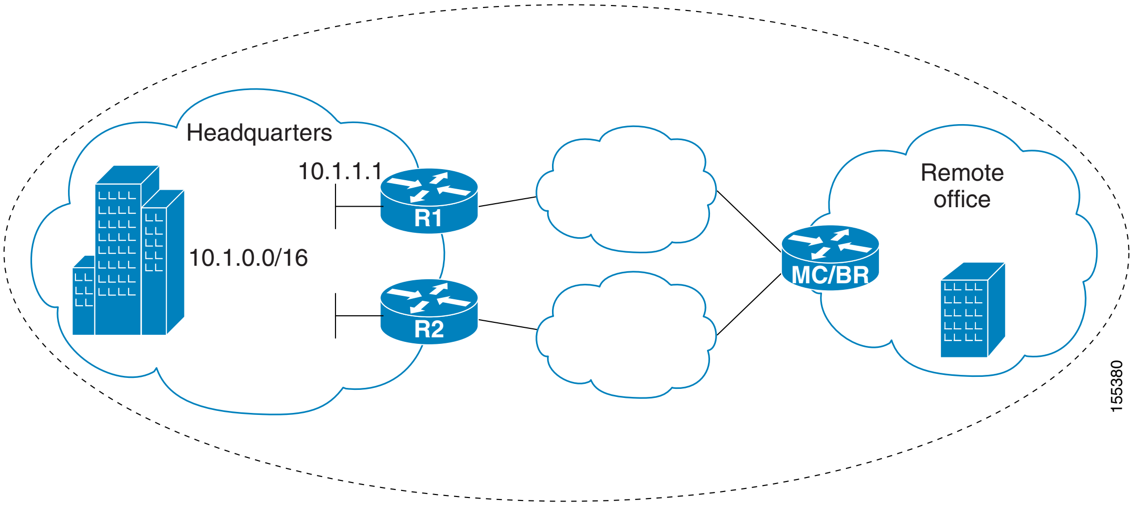
The following examples show both how to use an access list to identify only voice traffic to be optimized by OER and to use a prefix list to identify traffic that includes voice traffic to be optimized by OER.

- Optimizing Only Voice Traffic Using Active Probes, page 69
- Optimizing Traffic (Including Voice Traffic) Using Active Probes, page 70

Optimizing Only Voice Traffic Using Active Probes

Figure 6 shows that voice traffic originating at the remote office and terminating at the headquarters has to be optimized to select the best path out of the remote office network. Degradation in voice (traffic) quality is less likely to be introduced within the network, so probing the edge of the network gives a measurement that is close to probing the final destination.

Figure 6 OER Network Topology Optimizing Voice Traffic Using Active Probes



This configuration optimizes voice traffic to use the best performance path, whereas all other traffic destined to the same network—10.1.0.0/16—will follow the best path as indicated by a traditional routing protocol, for example BGP, that is configured on the device. As part of this optimization, OER will use policy based routing (PBR) to set the best exit link for voice traffic within a device.

The following configuration is performed on the edge router R1 in Figure 6 in the headquarters network to enable the IP SLAs Responder.

```
enable
configure terminal
 ip sla responder
 exit
```

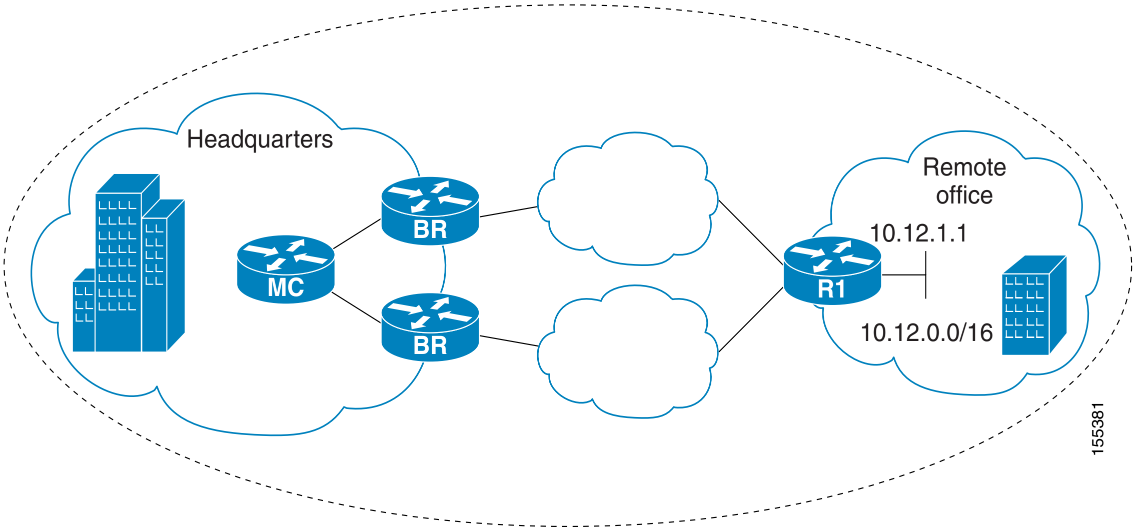
The following configuration is performed on the edge router MC/BR (which is both an OER master controller and border router) in Figure 6 in the remote office network to optimize voice traffic using active probes.

```
enable
configure terminal
ip access-list extended Voice_Traffic
 10 permit udp any 10.1.0.0 0.0.255.255 range 16384 32767
 exit
oer-map Voice_MAP 10
 match ip address access-list Voice_Traffic
 set active-probe jitter 10.1.1.1 target-port 1025 codec g711alaw
 set delay threshold 300
 set mos threshold 3.76 percent 30
 set jitter threshold 15
 set loss relative 5
 resolve mos priority 1
 resolve jitter priority 2
 resolve delay priority 3
 resolve loss priority 4
```

Optimizing Traffic (Including Voice Traffic) Using Active Probes

Figure 7 shows that traffic originating in the headquarters network and destined for the remote office network has to be optimized based on voice traffic metrics. Voice traffic is one of the most important traffic classes that travel from the headquarters to the remote office network, so the voice traffic must be prioritized to be optimized. Degradation in voice packet quality is less likely to be introduced within the network, so probing the edge of the network gives a measurement that is close to probing the final destination.

Figure 7 OER Network Topology for Optimizing All Traffic Using Active Probes



This configuration optimizes all traffic, including voice traffic, destined for the10.12.0.0/16 network. The OER optimization is based on the measurement of voice performance metrics with threshold values using active probes. As part of the optimization, OER will introduce a BGP or a static route into the headquarters network. For more details about BGP and static route optimization, see the Using OER to Control Traffic Classes and Verify the Route Control Changes module.

The following configuration is performed on router R1 in Figure 7 in the remote office network to enable the IP SLAs Responder.

```
enable
configure terminal
 ip sla responder
 exit
```

The following configuration is performed on one of the BR routers in Figure 7 in the headquarters network to optimize all traffic (including voice traffic) using active probes.

```
enable
configure terminal
 ip prefix-list All_Traffic_Prefix permit 10.12.0.0/16
 oer-map Traffic_MAP 10
 match ip address prefix-list All_Traffic_Prefix
 set active-probe jitter 10.12.1.1 target-port 1025 codec g711alaw
! port 1025 for the target probe is an example.
 set delay threshold 300
 set mos threshold 3.76 percent 30
 set jitter threshold 15
 set loss relative 5
 resolve mos priority 1
 resolve jitter priority 2
 resolve delay priority 3
 resolve loss priority 4
```

Where to Go Next

This module covered the OER apply policy phase and it has assumed that you started with the "Cisco IOS Optimized Edge Routing Overview" and the "Setting Up OER Network Components" module. The apply policy phase is the third phase in the OER performance loop. To learn more about the other OER phases, read through the other modules in the following list:

- Using OER to Profile the Traffic Classes
- Measuring the Traffic Class Performance and Link Utilization Using OER
- Configuring and Applying OER Policies
- Using OER to Control Traffic Classes and Verify the Route Control Changes

After you understand the various OER phases, review the OER solutions modules that are listed under "Related Documents" section on page 72.

Additional References

The following sections provide references related to configuring and applying OER policies.

Related Documents

Related Topic	Document Title
Cisco OER technology overview	"Cisco IOS Optimized Edge Routing Overview" module
Concepts and configuration tasks required to set up OER network components.	"Setting Up OER Network Components" module
OER solution module: voice traffic optimization using OER active probes.	"OER Voice Traffic Optimization Using Active Probes" module
OER solution module: configuring VPN IPsec/GRE tunnel interfaces as OER-managed exit links.	"Configuring VPN IPsec/GRE Tunnel Interfaces As OER-Managed Exit Links" module
Cisco OER commands: complete command syntax, command mode, command history, defaults, usage guidelines and examples	*Cisco IOS Optimized Edge Routing Command Reference*
IP Routing Protocol commands	*Cisco IOS IP Routing Protocols Command Reference*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Configuring and Applying OER Policies

Table 1 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.3(8)T, 12.2(33)SRB, or a later release appear in the table.

For information on a feature in this technology that is not documented here, see the "Cisco IOS Optimized Edge Routing Feature Roadmap."

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Configuring and Applying OER Policies

Feature Name	Releases	Feature Configuration Information
Optimized Edge Routing	12.3(8)T 12.2(33)SRB	OER was introduced.
OER Support for Policy-Rules Configuration	12.3(11)T 12.2(33)SRB	The OER Support for Policy-Rules Configuration feature introduced the capability to select an OER map and apply the configuration under OER master controller configuration mode, providing an improved method to switch between predefined OER maps. The following sections provide information about this feature: • Policy Rules Configuration to Apply an OER Policy, page 13 • Configuring Policy Rules for OER Maps, page 35 • Configuring Policy Rules for OER Maps: Example, page 66 The following commands were introduced or modified by this feature: **policy-rules**.

Table 1 Feature Information for Configuring and Applying OER Policies (continued)

Feature Name	Releases	Feature Configuration Information
OER Support for Cost-Based Optimization	12.3(14)T 12.2(33)SRB	The OER Support for Cost-Based Optimization feature introduced the capability to configure exit link policies based monetary cost and the capability to configure traceroute probes to determine prefix characteristics on a hop-by-hop basis. The following sections provide information about this feature: • OER Link Policies, page 6 • Configuring OER Cost-Based Policies, page 48 • Configuring OER Cost-Based Policies: Example, page 68 The following commands were introduced or modified by this feature: **cost-minimization**, **debug oer master cost-minimization**, **show oer master cost-minimization**.
OER Voice Traffic Optimization	12.4(6)T 12.2(33)SRB	The OER Voice Traffic Optimization feature introduced support for outbound optimization of voice traffic based on the voice metrics, jitter and Mean Opinion Score (MOS). Jitter and MOS are important quantitative quality metrics for voice traffic and these voice metrics are measured using OER active probes. The following sections provide information about this feature: • OER Traffic Class Performance Policies, page 5 • Configuring OER Voice Traffic Optimization Using Active Probes, page 54 • Configuring OER Voice Traffic Optimization Using Active Probes: Examples, page 69 The following commands were introduced or modified by this feature: **active-probe**, **jitter**, **mos**, **resolve**, **set jitter**, **set mos**, **set probe**, **set resolve**, **show oer master active-probes**, **show oer master policy**, **show oer master prefix**.

Table 1 Feature Information for Configuring and Applying OER Policies (continued)

Feature Name	Releases	Feature Configuration Information
OER BGP Inbound Optimization	12.4(9)T 12.2(33)SRB	OER BGP inbound optimization supports best entrance selection for traffic that originates from prefixes outside an autonomous system destined for prefixes inside the autonomous system. External BGP (eBGP) advertisements from an autonomous system to an Internet service provider (ISP) can influence the entrance path for traffic entering the network. OER uses eBGP advertisements to manipulate the best entrance selection. The following sections provide information about this feature: • Configuring and Applying an OER Policy to Learned Inside Prefixes, page 29 • Configuring and Applying an OER Policy to Configured Inside Prefixes, page 32 • Configuring and Applying an OER Policy to Learned Inside Prefixes: Example, page 65 • Configuring and Applying an OER Policy to Configured Inside Prefixes: Example, page 66 The following commands were introduced or modified by this feature: **clear oer master prefix**, **downgrade bgp**, **inside bgp**, **match ip address (OER)**, **match oer learn**, **max range receive**, **maximum utilization receive**, **show oer master prefix**.

Table 1 Feature Information for Configuring and Applying OER Policies (continued)

Feature Name	Releases	Feature Configuration Information
OER DSCP Monitoring	12.4(9)T 12.2(33)SRB	OER DSCP Monitoring introduced automatic learning of traffic classes based on protocol, port numbers, and DSCP value. Traffic classes can be defined by a combination of keys comprising of protocol, port numbers, and DSCP values, with the ability to filter out traffic that is not required, and the ability to aggregate the traffic in which you are interested. Information such as protocol, port number, and DSCP information is now sent to the master controller database in addition to the prefix information. The new functionality allows OER to both actively and passively monitor application traffic. The following sections provide information about this feature: • OER Traffic Class Performance Policies, page 5 • OER Policy Application, page 12 • Configuring and Applying an OER Policy to Configured Traffic Classes, page 19 • Configuring and Applying an OER Policy to Configured Traffic Classes: Example, page 65 The following commands were introduced or modified by this feature: **show oer border passive applications**, **show oer border passive cache**, **show oer border passive learn**, **show oer master appl**, **traffic-class aggregation**, **traffic-class filter**, and **traffic-class keys**.
Performance Routing - Link Groups	12.4(15)T	The Performance Routing - Link Groups feature introduces the ability to define a group of exit links as a preferred set of links, or a fallback set of links for OER to use when optimizing traffic classes specified in an OER policy. The following sections provide information about this feature: • Performance Routing Link Grouping, page 9 • Implementing Performance Routing Link Groups, page 42 • Implementing Performance Routing Link Groups: Example, page 67 The following commands were introduced or modified by this feature: **link-group**, **set link-group**, and **show oer master link-group**.

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

Using OER to Control Traffic Classes and Verify the Route Control Changes

First Published: January 29, 2007
Last Updated: March 30, 2008

This module describes the Optimized Edge Routing (OER) control and verify phases. During the previous OER phases OER operates, by default, under observe mode. In observe mode, the master controller (MC) coordinates performance information from the border routers and makes policy decisions, but no route control action is taken. When the OER control mode is enabled, the master controller coordinates information from the borders routers in the same way as observe mode, but commands are sent back to the border routers to alter routing in the OER managed network to implement the policy decisions. OER controls the traffic defined in a traffic class through entrance and exit links selected on the basis of the default or user-defined policies. After controlling traffic, the last phase of the OER performance loop is to verify that the OER control actions implement changes to the flow of traffic and that the performance of the traffic class or exit does move to an in-policy state. OER troubleshooting using traceroute reporting is also documented in this module.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Using OER to Control Traffic Classes and Verify the Route Control Changes" section on page 31.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Contents

Prerequisites for Using OER to Control Traffic Classes and Verify the Route Control Changes

- Before implementing the OER policy phase, you need to understand an overview of how OER works and how to set up OER network components. See the "Cisco IOS Optimized Edge Routing Overview" and "Setting Up OER Network Components" modules for more details. If you are following the OER performance loop, the OER learn, measure, and policy phases precede this phase. See the "Where to Go Next" section on page 29 for more details.
- Either routing protocol peering must be established on your network or static routing must be configured before route control mode is enabled.

  If you have configured internal Border Gateway Protocol (iBGP) on the border routers, BGP peering must be either established and consistently applied throughout your network or redistributed into an Interior Gateway Protocol (IGP). The following IGPs are supported: Enhanced Interior Gateway Routing Protocol (EIGRP), Open Shortest Path First (OSPF), Intermediate System-to-Intermediate System (IS-IS), or Routing Information Protocol (RIP).

  If an IGP is deployed in your network, static route redistribution must be configured with the **redistribute** command unless iBGP is configured. IGP or static routing should also be applied consistently throughout an OER-managed network; the border router should have a consistent view of the network.

Caution Caution must be applied when redistributing OER static routes into an IGP. The routes injected by OER may be more specific than routes in the IGP, and it will appear as if the OER border router is originating these routes. To avoid routing loops, the redistributed OER static routes should never be advertised over a WAN by an OER border router or any other router. Route filtering and stub network configuration can be used to prevent advertising the OER static routes. If the OER static routes are redistributed to routers terminating the OER external interfaces, routing loops may occur.

For more details about configuring routing protocol peering or redistribution between border routers and peer routers, see the "Setting Up OER Network Components" module.

Information About Using OER to Control Traffic Classes and Verify the Network Performance

To configure OER to control traffic class and verify the network performance, you should understand the following concepts:

OER Control Phase Overview

After profiling the traffic classes during the OER learn phase, measuring the performance metrics of the traffic classes during the measure phase, and using network policies to map the measured performance metrics of traffic class entries in the Monitored Traffic Class (MTC) list against well-known or configured thresholds to determine if the traffic is meeting specified levels of service in the policy phase, the next step in the OER performance loop is the OER control phase.

OER, by default, operates in an observation mode and the documentation for the OER learn, measure, and apply policy phases assumes that OER is in the observe mode. In observe mode, the master controller monitors traffic classes and exit links based on default and user-defined policies and then reports the status of the network including out-of-policy (OOP) events and the decisions that should be made, but does not implement any changes. The OER control phase operates in control mode, not observe mode, and control mode must be explicitly configured using the **mode route control** command. In control mode, the master controller coordinates information from the borders routers in the same way as observe mode, but commands are sent back to the border routers to alter routing in the OER managed network to implement the policy decisions.

OER initiates route changes when one of the following occurs:

- A traffic class goes OOP.
- An exit link goes OOP.
- The periodic timer expires and the select exit mode is configured as select best mode.

During the OER control phase, the master controller continues to monitor in-policy traffic classes that conform to the desired performance characteristics, to ensure that they remain in-policy. Changes are only implemented for OOP traffic classes and exits in order to bring them in-policy. To achieve the desired level of performance in your network, you must be aware of the configuration options that can affect the policy decisions made by the master controller. The following options, if configured, can influence the behavior of OER when making routing and policy decisions:

- Backoff timer—The backoff timer associated with each traffic class is configured with minimum and maximum values. If the select exit best option is configured and a prefix associated with a traffic class is OOP on all available exits, then the best available exit will be selected for a period of time, in seconds, as specified for the backoff timer. Each time the backoff timer expires and OER fails to discover an in-policy exit, the backoff interval is increased by a specified step (if configured) or a minimum number of seconds that increases up to a maximum number of seconds. If the backoff timer expires and the current exit is in-policy, the backoff timer is reset to the minimum number of seconds. If the select exit good option is configured, and a traffic class goes OOP and cannot be controlled, OER transitions the traffic class to a default state and the backoff timer is started. When the backoff timer expires, OER attempts to find the first in-policy exit, but if the traffic class is still OOP on all exits and transitions back to the default state, the backoff timer will be incremented.

- Holddown timer—The holddown setting specifies the minimum period of time that a new exit must be used before an alternative exit can be selected. The exception to this rule is when a traffic class is determined to be unreachable while it is still in the holddown state, in which case the prefix is immediately moved to the first exit through which the traffic class is reachable. Holddown is used to reduce route flapping.
- Select exit—In passive monitoring mode, the configuration that specifies whether the exit selection is good or best specifies the algorithm used to choose an alternative exit for a prefix. If select good is configured, the first exit that conforms to the policy is selected as the new exit. If OER does not find an in-policy exit for a traffic class when the select good is operational, OER transitions the traffic class to an uncontrolled (default) state. If select best is configured, information is collected from all exits, and the best one is selected even though the best exit may not be in-policy. In active monitoring mode, if select exit is configured, OER selects the best exit even if the exit is OOP. If the exit is OOP, OER moves the traffic class to OOP. If select good is selected in active mode, and the traffic class is OOP on all the exits, OER transitions the traffic class to an uncontrolled state.
- Periodic timer—When the periodic timer expires, the master controller evaluates the current path of the traffic class based on default or user-defined policies. OER will select either the best exit or the first in-policy exit depending on the select exit configuration.

The backoff and holddown timers can be used to provide dampening in an OER-managed network. Dampening generally refers to the attempt to find a compromise between quick reactions to network events versus the unwanted network churn that can occur when reacting to multiple events happening within a short time period. The main purpose for dampening is to allow the software to react quickly to initial events, while giving the network time to adjust to the changes before initiating any further actions. In OER, after a policy decision for a traffic class or link has caused routing changes, the same traffic class or link cannot cause further changes until a set period of time has expired.

Another configuration issue to consider when deploying OER is that if aggressive delay or loss policies are defined, and the exit links are also seriously over-subscribed, it is possible that OER will find it impossible to bring a traffic class in-policy. In this case, the master controller will either choose the link that most closely conforms to the performance policy, even though the traffic class still remains OOP, or it will remove the prefix from OER control. OER is designed to allow you to make the best use of available bandwidth, but it does not solve the problem of over-subscribed bandwidth.

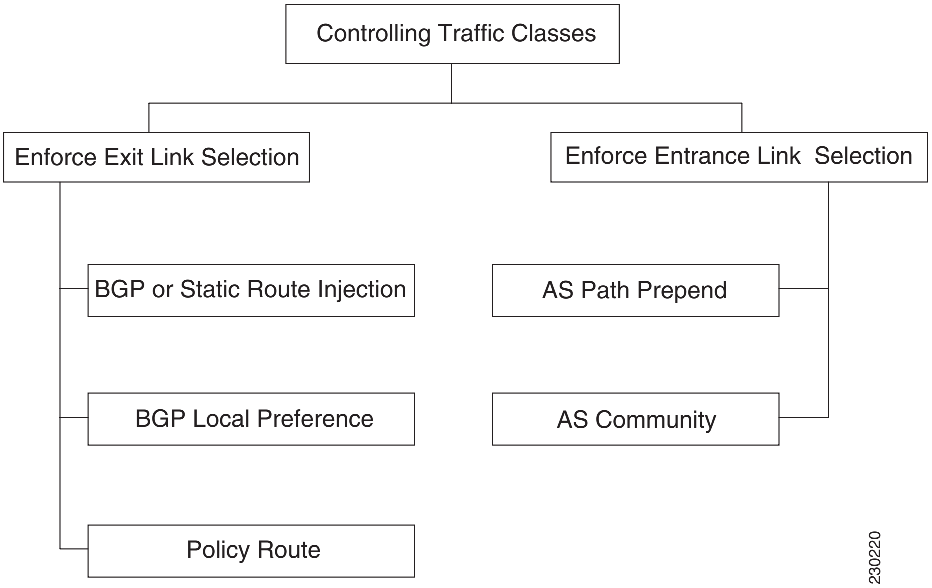
After OER control mode is enabled, and configuration options are considered, the next step is to review the traffic class control techniques employed by OER.

OER Traffic Class Control Techniques

After the OER master controller has determined that it needs to take some action involving an OOP traffic class or exit link, there are a number of techniques that can be used to alter the routing metrics or BGP attributes to influence traffic to use a different link. If the traffic associated with the traffic class is defined only by a prefix then a traditional routing control mechanism such as introducing a BGP route or a static route can be deployed. This control is network wide after redistribution because a prefix introduced into the routing protocol with a better metric will attract traffic for that prefix towards a border router. If the traffic associated with the traffic class is defined by a prefix and other matching criteria for the packet (application traffic, for example), then traditional routing cannot be employed to control the application traffic. In this situation, the control becomes device specific and not network specific. This device specific control is implemented by OER using policy-based routing (PBR) functionality. If the traffic in this scenario has to be routed out to a different device, the remote border router should be a single hop away or a tunnel interface that makes the remote border router look like a single hop.

Figure 1 shows the various traffic class control techniques grouped by exit or entrance link selection. In the initial OER releases, only exit link selection could be controlled. In Cisco IOS Release 12.4(9)T, 12.2(33)SRB, and later releases, the ability to control entrance selection was introduced.

Figure 1 Controlling Traffic Class Techniques


Controlling Traffic Classes
Enforce Exit Link Selection
Enforce Entrance Link Selection
BGP or Static Route Injection
AS Path Prepend
BGP Local Preference
AS Community
Policy Route
230220

For more details about the different traffic class control techniques that can be implemented, see the following sections:

- OER Exit Link Selection Control Techniques, page 5
- OER Entrance Link Selection Control Techniques, page 7

OER Exit Link Selection Control Techniques

To enforce an exit link selection, OER offers the following methods:

- Static Route Injection, page 5
- BGP Control Techniques, page 6
- Policy Route, page 6

Static Route Injection

An OER master controller can enforce the use of a particular border router as the preferred exit link for a traffic class by injecting temporary static routes. These static routes exist only in the memory of the router, and are intentionally not saved to the permanent configuration. There are a few different methods that the master controller can use to inject static routes on the border routers. Existing static routes can be overwritten with new static routes, which have a better routing metric. If a default route, or even a less specific route, exists on the border router, the master controller can add a specific static route for the monitored traffic classes, which will be preferred to the existing default route. Finally, the master controller can also use something known as split prefixes.

A split prefix refers to the addition of a more specific route, which will be preferred over a less specific route. For example, if the border router already has a route of 10.10.10.0/24, adding a static route of 10.10.10.128/25 will also cause the addresses 10.10.10.129-10.10.10.254 to be forwarded using the newly injected route. If OER has been configured to monitor a subset of a larger network, it will add an appropriate route to the existing routing table. OER can use split prefixes to redirect subsets of an existing prefix to a more optimal exit link, and can use split prefixes for both internal BGP (iBGP) and static routes.

OER will never inject a route where one does not already exist in the routing protocol table. Before injecting a route of a particular type, OER will verify that a route exists in the BGP or static table that includes the prefix and points to the exit link. This route may be a default route.

BGP Control Techniques

OER uses two BGP techniques to enforce the best exit path; injecting a BGP route, or modifying the BGP local preference attribute.

If the traffic associated with the traffic class is defined only by a prefix, the master controller can instruct a border router to inject a BGP route into the BGP table to influence traffic to use a different link. All OER injected routes remain local to an autonomous system, and these injected routes are never shared with external BGP peers. As a safeguard to ensure this behavior, when OER injects a BGP route, it will set the no-export community on it. This is done automatically, and does not require any user configuration. However, because these routes now have a special marking, some extra configuration is required to allow the information to be shared with internal BGP peers. For each iBGP peer, the send community configuration must be specified. Although the border routers know about the best exit for the injected route, it may also be necessary to redistribute this information further into the network. For more details about redistribution in an OER-managed network, see the "Setting Up OER Network Components" module.

OER also uses BGP local preference to control traffic classes. BGP local preference (Local_Pref) is a discretionary attribute applied to a BGP prefix to specify the degree of preference for that route during route selection. The Local_Pref is a value applied to a BGP prefix, and a higher Local_Pref value causes a route to be preferred over an equivalent route. The master controller instructs one of the border routers to apply the Local_Pref attribute to a prefix or set of prefixes associated with a traffic class. The border router then shares the Local_Pref value with all of its internal BGP peers. Local_Pref is a locally significant value within an autonomous system, but it is never shared with external BGP peers. Once the iBGP reconvergence is complete, the router with the highest Local_Pref for the prefix will become the exit link from the network.

Note If a local preference value of 5000 or higher has been configured for default BGP routing, you should configure a higher BGP local preference value in OER using the **mode** command in OER master controller configuration mode.

Policy Route

In Cisco IOS Release 12.4(2)T, 12.2(33)SRB, and later releases, OER can control application traffic using policy-based routing. Application traffic traveling through a particular OER border router can be identified by matching traffic defined in an OER map as part of an OER policy. The **match ip address** (OER) command was enhanced to support extended ACLs. The extended ACL is referenced in an OER map, and a single match clause can be configured for each OER map sequence. Set clauses are configured to apply independent OER policies to the matched traffic, which is a subset of a monitored prefix. The OER policy is applied to all border routers to enforce policy routing for the application. Matched traffic is policy routed through the OER external interface that conforms to policy parameters.

In Cisco IOS Release 12.4(9)T, 12.2(33)SRB, and later releases, the ability to use DSCP values, as well as prefixes, port numbers, and protocols, to identify and control application traffic was introduced. DSCP values, protocols, and port numbers are now sent by the border routers to the master controller for inclusion in the MTC list.

OER Entrance Link Selection Control Techniques

In Cisco IOS Release 12.4(9)T, 12.2(33)SRB, and later releases, the ability to influence inbound traffic was introduced with the OER BGP inbound optimization feature. A network advertises reachability of its inside prefixes to the Internet using eBGP advertisements to its ISPs. If the same prefix is advertised to more than one ISP, then the network is multihoming. OER BGP inbound optimization works best with multihomed networks, but it can also be used with a network that has multiple connections to the same ISP. To implement BGP inbound optimization, OER manipulates eBGP advertisements to influence the best entrance selection for traffic bound for inside prefixes. The benefit of implementing the best entrance selection is limited to a network that has more than one ISP connection.

To enforce an entrance link selection, OER offers the following methods:

- BGP Autonomous System Number Prepend, page 7
- BGP Autonomous System Number Community Prepend, page 7

BGP Autonomous System Number Prepend

After OER selects a best entrance for an inside prefix, extra autonomous system hops (up to a maximum of six) are prepended to the inside prefix BGP advertisement over the other entrances. The extra autonomous system hops on the other entrances increase the probability that the best entrance will be used for the inside prefix. This is the default method OER uses to control an inside prefix, and no user configuration is required.

BGP Autonomous System Number Community Prepend

After OER selects a best entrance for an inside prefix, a BGP prepend community is attached to the inside prefix BGP advertisement from the network to another autonomous system such as an ISP. The BGP prepend community will increase the number of autonomous system hops in the advertisement of the inside prefix from the ISP to its peers. Autonomous system prepend BGP community is the preferred method to be used for OER BGP inbound optimization because there is no risk of the local ISP filtering the extra autonomous system hops. There are some issues, for example, not all ISPs support the BGP prepend community, ISP policies may ignore or modify the autonomous system hops, and a transit ISP may filter the autonomous system path. If you use this method of inbound optimization and a change is made to an autonomous system, you must issue an outbound reconfiguration using the **clear ip bgp** command.

OER Verify Phase

The last phase of the OER performance loop is to verify that the actions taken during the OER control phase control actually change the flow of traffic and that the performance of the traffic class or link does move to an in-policy state. OER uses NetFlow to automatically verify the route control. The master controller expects a Netflow update for the traffic class from the new link interface and ignores Netflow updates from the previous path. If a Netflow update does not appear after two minutes, the master controller moves the traffic class into the default state. A traffic class is placed in the default state when it is not under OER control.

In addition to the NetFlow verification used by OER, there are two other methods you can use to verify that OER has initiated changes in the network:

- Syslog report—The logging command can be configured to notify you of all the main OER state changes, and a syslog report can be run to confirm that OER changes have occurred. The master controller is expecting bidirectional traffic, and a syslog report delimited for the specified prefix associated with the traffic class can confirm this.
- OER show commands—OER show commands can be used to verify that network changes have occurred and that traffic classes are in-policy. Use the **show oer master prefix** command to display the status of monitored prefixes. The output from this command includes information about the current exit interface, prefix delay, egress and ingress interface bandwidth, and path information sourced from a specified border router. Use the **show oer border routes** command to display information about OER controlled routes on a border router. This command can display information about BGP or static routes.

OER Troubleshooting Using Traceroute Reporting

Although OER provides the ability to diagnose issues using **syslog** and **debug** command-line interface (CLI) commands, support for traceroute reporting was introduced in Cisco IOS Release 12.3(14)T and 12.2(33)SRB. Using traceroute reporting, OER reports traffic class performance by determining the delay on a hop-by-hop basis using traceroute probes.

Prior to traceroute reporting there was no method for measuring the delay per hop for situations such as an unexpected round trip delay value being reported for a traffic class on an exit link. OER uses UDP traceroutes to collect per-hop delay statistics. A traceroute is defined as tracing the route to the device with the given IP address or the hostname and is useful in detecting the location of a problem that exists in the path to the device. Although traditional UDP-based traceroutes are used by default, OER can be configured to send TCP SYN packets to specific ports that may be permitted through a firewall.

Traceroute reporting is configured on the master controller. Traceroute probes are sourced from the border router exit. This feature allows you to monitor traffic class performance on a hop-by-hop basis. When traceroute reporting is enabled, the autonomous system number, the IP address, and delay measurements are gathered for each hop from the probe source to the target prefix. By default, traceroute probes are sent only when the traffic class goes OOP. TCP-based traceroutes can be configured manually and the time interval between traceroute probes can be modified. By default, per-hop delay reporting is not enabled.

Traceroute probes are configured using the following methods:

- Periodic—A traceroute probe is triggered for each new probe cycle. The probe is sourced from the current exit of the traffic class when the option to probe only one exit is selected. If the option to probe all exits is selected, the traceroute probe is sourced from all available exits.
- Policy based—A traceroute probe is triggered automatically when a traffic class goes into an out-of-policy state. Traceroute reporting can be enabled for all traffic classes specified in the match clause of an OER map. Policy based traceroute reporting stops when the traffic class returns to an in-policy state.

On demand—A trace route probe can be triggered on an on demand basis when periodic traceroute reporting is not required, or the per-hop statistics are not required for all paths. Using optional keywords and arguments of the **show oer master prefix** command, you can start traceroute reporting for a specific traffic class on a specific path, or all paths.

How to Use OER to Control Traffic Classes and Verify the Route Control Changes

This section contains tasks to configure OER to control traffic classes and verify the route control changes. The first task in the following list is enables OER route control mode to control routes globally. After OER route control mode is enabled, the master controller can either influence an existing route or, if there is no exact match, OER can inject a BGP or static route if a less specific prefix is found. Influencing an existing route or injecting a new route are used to control the traffic defined by a traffic class when it goes out-of-policy. If route injection is used, the master controller automatically tries to inject a BGP route first, but if a BGP parent route does not exist, a static route is injected. The next two tasks allow you to set a BGP local preference value or a static tag value for routes used by OER to control traffic classes. The fourth task configures OER to control application traffic and uses an OER map to enable control only of specified routes. The fifth task configures load balancing of entrance links using inside prefix traffic, and the sixth task shows how to verify the route control changes that OER has implemented in the network. The last task configures traceroute reporting for troubleshooting purposes.

- Enabling OER Route Control Mode, page 9
- Setting a Tag Value for Injected OER Static Routes, page 10
- Setting a BGP Local Preference Value for OER Controlled BGP Routes, page 11
- Controlling Application Traffic, page 13
- Enforcing Entrance Link Selection with Load Balancing for an Inside Prefix, page 16
- Manually Verifying the OER Route Control Changes, page 22
- Configuring Traceroute Reporting, page 24

Enabling OER Route Control Mode

Perform this task on the master controller to enable the OER route control mode. In this task, route control is enabled globally for all subsequent policies. After enabling global route control, most of the policy tasks in the "Configuring and Applying OER Policies" module can be performed and OER will control any of the traffic classes or links that are OOP. To control the traffic defined by a traffic class when it goes out-of-policy, the master controller can either influence an existing route or, if there is no exact match, OER can inject a BGP or static route if a less specific prefix is found. If route injection is used, the master controller automatically tries to inject a BGP route first, but if a parent route does not exist in the BGP table, a static route is injected.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **oer master**
4. **mode route control**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**oer master** **Example:** `Router(config)# oer master`	Enters OER master controller configuration mode to configure a router as a master controller and to configure global operations and policies.
Step 4	**mode route control** **Example:** `Router(config-oer-mc)# mode route control`	Configures an operational mode on a master controller. • The **route** and **control** keywords enable route control mode. In control mode, the master controller analyzes monitored traffic classes and implements changes based on policy parameters. **Note** Only the syntax applicable to this task is shown. For more details, see the *Cisco IOS Optimized Edge Routing Command Reference*.

Step 5	**end**
	Example:
	`Router(config-oer-mc)# end`
		Exits OER master controller configuration mode and returns to privileged EXEC mode.

Setting a Tag Value for Injected OER Static Routes

Perform this task on the master controller to set a tag value for an injected static route to allow the routes to be uniquely identified. A static route may be injected by OER to control the traffic defined by a traffic class when it goes out-of-policy. By default, OER uses a tag value of 5000 for injected static routes, but OER offers the ability to configure a different value. In this task, the OER route control mode is configured globally with the **mode** command in OER master controller configuration mode and any injected static routes will be tagged with a value of 7000. Using the static tag value, OER routes can be redistributed or filtered using route maps.

You can also configure this task to apply to specific prefixes using the **set mode** command in OER map configuration mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **oer master**
4. **mode route control**

5. **mode route metric** {**bgp local-pref** *preference* | **static tag** *value*}
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**oer master** **Example:** `Router(config)# oer master`	Enters OER master controller configuration mode to configure a router as a master controller and to configure global operations and policies.
Step 4	**mode route control** **Example:** `Router(config-oer-mc)# mode route control`	Configures the OER route control mode on a master controller. • The **route** and **control** keywords enable route control mode. In control mode, the master controller analyzes monitored traffic classes and implements changes based on policy parameters. **Note** Only the syntax applicable to this task is shown. For more details, see the *Cisco IOS Optimized Edge Routing Command Reference*.

Step 5	**mode route metric** {**bgp local-pref** *preference* | **static tag** *value*}
	Example:
	`Router(config-oer-mc)# mode route metric static tag 7000`
		Sets a BGP local preference value or a static tag value for injected BGP or static routes.
		• Use the **static** and **tag** keywords to apply a tag to a static route under OER control. The *value* argument is a number from 1 to 65535.
		Note Only the syntax applicable to this task is shown. For more details, see the *Cisco IOS Optimized Edge Routing Command Reference*.

Step 6	**end**
	Example:
	`Router(config-oer-mc)# end`
		Exits OER master controller configuration mode and returns to privileged EXEC mode.

Setting a BGP Local Preference Value for OER Controlled BGP Routes

Perform this task on the master controller to set a BGP local preference attribute value. OER uses the BGP local preference (Local_Pref) value to influence the BGP best path selection on internal BGP (iBGP) neighbors as a method of enforcing exit link selection. By default, OER uses a BGP Local_Pref value of 5000, but OER offers the ability to configure a different value. If a Local_Pref value of 5000 or higher has been configured for default BGP routing, you should configure a higher BGP Local_Pref value because a higher Local_Pref value causes a route to be preferred over an equivalent route. In this task, route control is enabled for traffic matching a prefix list and the BGP local preference value of 40000 is set.

All OER injected routes remain local to an autonomous system, and these injected routes are never shared with external BGP peers. As a safeguard to ensure this behavior, when OER injects a BGP route, it will set the no-export community on it. This is done automatically, and does not require any user configuration. However, because these routes now have a community marking, the send community configuration must be specified for each iBGP peer to allow the information to be shared with internal BGP peers. Although the border routers know about the best exit for the injected route, it may also be necessary to redistribute this information further into the network.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **oer-map** *map-name sequence-number*
4. **match ip address prefix-list** *prefix-list-name*
5. **set mode route control**
6. **set mode metric** {**bgp local-pref** *preference* | **static tag** *value*}
7. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`oer-map`** *`map-name sequence-number`*
	Example:
	`Router(config)# oer-map MODE 10`
		Enters OER map configuration mode to configure an OER map to apply policies to selected IP prefixes.
		• *Only a single match clause can be configured for each OER map sequence.*
		• The example creates an OER map named MODE.

Step 4	**`match ip address prefix-list`** *`prefix-list-name`*
	Example:
	`Router(config-oer-map)# match ip address prefix-list RED`
		Creates a prefix list match clause entry in an OER map to apply OER policies or creates a match clause entry in an oer-map to match OER learned prefixes.
		• The **match ip address** command supports IP prefix lists only.
		• The example configures the prefix list named RED as match criteria in an OER map.

	Command or Action	Purpose

Step 5	**`set mode route control`**
	Example:
	`Router(config-oer-map)# set mode route control`
		Creates a set clause entry to configure route control for matched traffic.
		• Use the **route** and **control** keywords to enable route control mode. In control mode, the master controller analyzes monitored traffic classes and implements changes based on policy parameters.
		• In this example, a set clause that enables OER control mode is created.
		Note Only the syntax applicable to this task is shown. For more details, see the *Cisco IOS Optimized Edge Routing Command Reference*.

Step 6	**`set mode route metric`** `{`**`bgp local-pref`** *`preference`* `|` **`static tag`** *`value`*`}`
	Example:
	`Router(config-oer-map)# set mode route metric bgp local-pref 40000`
		Sets a BGP local preference value or a static tag value for injected BGP or static routes.
		• The **route** keyword enables route control mode. In control mode, the master controller analyzes monitored traffic classes and implements changes based on policy parameters.
		• Use the **bgp** and **local-pref** keywords to set the BGP local preference attribute for OER controlled routes. The *preference* argument is a number from 1 to 65535.
		Note Only the syntax applicable to this task is shown. For more details, see the *Cisco IOS Optimized Edge Routing Command Reference*.

Step 7	**`end`**
	Example:
	`Router(config-oer-map)# end`
		Exits OER map configuration mode, and return to privileged EXEC mode.

Controlling Application Traffic

Perform this task on a master controller to control application traffic. This task shows how to use policy-based routing (PBR) to allow OER to control specified application traffic classes. Application-aware policy routing was introduced in Cisco IOS Release 12.4(2)T and 12.2(33)SRB to configure application traffic that can be filtered with a permit statement in an extended IP access list.

Application traffic such as Telnet traffic is delay sensitive and long TCP delays can make Telnet sessions difficult to use. In this task, an extended IP access list is configured to permit Telnet traffic. An OER map is configured with an extended access list that references a match clause to match Telnet traffic that is sourced from the 192.168.1.0/24 network. OER route control is enabled and a delay policy is configured to ensure that Telnet traffic is sent out through exit links with a response time that is equal to, or less than, 30 milliseconds. The configuration is verified with the **show oer master appl** command.

Note In Cisco IOS Release 12.4(9)T, 12.2(33)SRB, and later releases, the ability to use DSCP values, as well as prefixes, port numbers, and protocols, to identify and control application traffic was introduced.

Prerequisites

The master controller and border routers must be running Cisco IOS Release 12.4(2)T, 12.2(33)SRB, or later releases.

Restrictions

- Border routers must be single-hop peers. If the border routers are separated by more than one hop, you must configure any Cisco router between the border routers as an interim border router.
- Only named extended IP access lists are supported
- Application traffic optimization is supported in OER only over CEF switching paths

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list** {**standard** | **extended**} *access-list-name*}
4. [*sequence-number*] **permit** *protocol source source-wildcard destination destination-wildcard* [**option** *option-name*] [**precedence** *precedence*] [**tos** *tos*] [**ttl** *operator value*] [**log**] [**time-range** *time-range-name*] [**fragments**]
5. **exit**
6. **oer-map** *map-name sequence-number*
7. **match ip address** {**access-list** *name* | **prefix-list** *name*}
8. **set mode route control**
9. **set delay** {**relative** *percentage* | **threshold** *maximum*}
10. **set resolve** {**cost priority** *value* | **delay priority** *value* **variance** *percentage* | **loss priority** *value* **variance** *percentage* | **range priority** *value* | **utilization priority** *value* **variance** *percentage*}
11. **end**
12. **show oer master appl** [**access-list** *name*] [**detail**] | [**tcp** | **udp**] [*protocol-number*] [*min-port max-port*] [**dst** | **src**] [**detail** | **policy**]

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		- Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

	Command or Action	Purpose

Step 3	**ip access-list** {**standard** | **extended**} *access-list-name*}
	Example:
	`Router(config)# ip access-list extended TELNET_ACL`
		Creates an extended access list and enters extended access list configuration mode.
		• Only named access lists are supported.

Step 4	[*sequence-number*] **permit** *protocol source source-wildcard destination destination-wildcard* [**option** *option-name*] [**precedence** *precedence*] [**tos** *tos*] [**ttl** *operator value*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Router(config-ext-nacl)# permit tcp 192.168.1.0 0.0.0.255 any eq telnet`	Defines the extended access list. • Any protocol, port, or other IP packet header value can be specified. • The example permits Telnet traffic that is sourced from the 192.168.1.0/24 network.
Step 5	**exit** **Example:** `Router(config-ext-nacl)# exit`	Exits extended access list configuration mode, and returns to global configuration mode.
Step 6	**oer-map** *map-name sequence-number* **Example:** `Router(config# oer-map BLUE`	Enters oer-map configuration mode to configure an OER map.

Step 7	**match ip address** {*access-list name* | *prefix-list name*}
	Example:
	`Router(config-oer-map)# match ip address access-list TELNET`
		References an extended IP access list or IP prefix as match criteria in an OER map.
		• An extended IP access list is used to filter a subset of traffic from the monitored prefix.

Step 8	**set mode route control** **Example:** `Router(config-oer-map)# set mode route control`	Creates a set clause entry to configure route control for matched traffic. • In control mode, the master controller analyzes monitored prefixes and implements changes based on policy parameters. • In this example, a set clause that enables OER control mode is created. **Note** Only the syntax applicable to this task is shown. For more details, see the *Cisco IOS Optimized Edge Routing Command Reference*.

Step 9	set delay {**relative** *percentage* | **threshold** *maximum*}
	Example:
	`Router(config-oer-map)# set delay threshold 30`
		(Optional) Configures an OER map to configure OER to set the delay threshold.
		• This example configures a delay policy. However, other policies could be configured.
		• The delay threshold is set to 30 milliseconds for Telnet traffic.

	Command or Action	Purpose

Step 10	**set resolve** {**cost priority** *value* | **delay priority** *value* **variance** *percentage* | **loss priority** *value* **variance** *percentage* | **range priority** *value* | **utilization priority** *value* **variance** *percentage*}
	Example:
	`Router(config-oer-map)# set resolve delay priority 1 variance 20`
		(Optional) Configures an oer-map to set policy priority for overlapping policies.
		• The resolve policy configures delay policies to have the highest priority with a 20 percent variance.

Step 11	**end**
	Example:
	`Router(config-oer-map)# end`
		Exits oer-map configuration mode and returns to privileged EXEC mode.

Step 12	**show oer master appl** [**access-list** *name*] [**detail**] | [**tcp** | **udp**] [*protocol-number*] [*min-port max-port*] [**dst** | **src**] [**detail** | **policy**] **Example:** `Router# show oer master appl tcp 23 23 dst policy`	Displays information about applications monitored and controlled by an OER master controller.

Examples

The following example output from the **show oer master appl** command shows TCP application traffic filtered based on port 23 (Telnet):

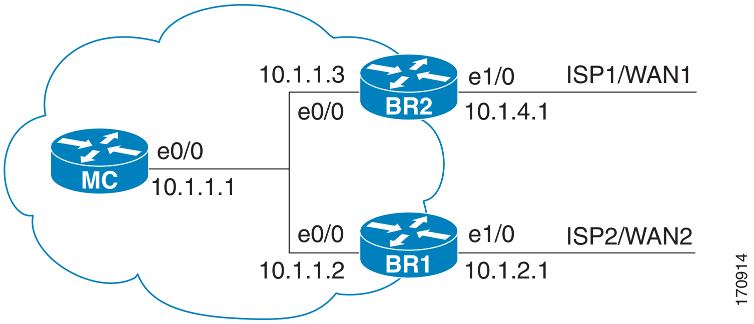
```
Router# show oer master appl tcp 23 23 dst policy

Prefix              Appl Prot       Port                 Port Type       Policy
-------------------------------------------------------------------------------
10.1.1.0/24        tcp             [23, 23]             src             10
```

Enforcing Entrance Link Selection with Load Balancing for an Inside Prefix

Perform this task on the master controller to enforce entrance link selection and load balancing for an inside prefix using a BGP autonomous system number community prepend. External BGP advertisements from the network to another autonomous system such as an ISP can influence the entrance link used for inbound traffic. OER can manipulate the best entrance by influencing the eBGP advertisement. In this task, after OER has selected the best entrance for an inside prefix, a BGP prepend community is attached to the inside prefix BGP advertisements from the other entrances that are not the OER-preferred entrances. The BGP prepend community will increase the number of autonomous system hops in the advertisement of the inside prefix from the ISP to its peers. Autonomous system BGP community prepend is the preferred method to be used for OER BGP inbound optimization because there is no risk of the local ISP filtering the extra autonomous system hops.

This task also shows how to configure a load balancing policy for traffic class flows over the border router entrance links. In this example, an inbound (receive) traffic utilization threshold policy and an inbound traffic utilization range policy are given priority when OER chooses the best link selection for inbound traffic classes. Best route selection for performance policies is disabled. The external Ethernet interfaces on border router 1 and border router 2—BR1 and BR2 in Figure 2—are both configured with a maximum inbound utilization threshold of 90 percent and a range of inbound utilization between the two links is set to 20 percent. After an external interface is configured for the border routers, OER automatically monitors the inbound traffic utilization of external links on a border router every 5 minutes. The inbound traffic utilization is reported back to the master controller and, if the inbound traffic utilization exceeds 90 percent, OER selects another link for inbound traffic classes on that link. To complete the load balancing, the utilization range between the two entrance links must not be greater than 20 percent, otherwise OER will move some of the traffic classes from one entrance link to another to balance the incoming traffic load between the two entrance links.

Figure 2 Network diagram for OER Entrance Link Load Balancing



Note Policies applied in an OER map do not override global policy configurations.

Prerequisites

The master controller and border routers must be running Cisco IOS Release 12.4(9)T, 12.2(33)SRB, or later releases.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **oer master**
4. **mode select-exit** {**best** | **good**}
5. **resolve range priority** *value*
6. **resolve utilization priority** *value* **variance** *percentage*
7. **no resolve delay**
8. **no resolve loss**
9. **max range receive percent** *percentage*
10. **border** *ip-address* [**key-chain** *key-chain-name*]
11. **interface** *type number* **external**
12. **maximum utilization receive** {**absolute** *kbps* | **percent** *percentage*}
13. **downgrade bgp community** *community-number*
14. **exit**
15. Repeat Step 14 twice to return to global configuration mode

16. **oer-map** *map-name sequence-number*
17. **match oer learn** {**delay** | **inside** | **throughput**}
18. **set delay** {**relative** *percentage* | **threshold** *maximum*}
19. **set mode route control**
20. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`oer master`**
	Example:
	`Router(config)# oer master`
		Enters OER master controller configuration mode to configure a router as a master controller.
		• A master controller and border router process can be enabled on the same router (for example, in a network that has a single router with two exit links to different service providers).
		Note Only the syntax used in this context is displayed. For more details, see the *Cisco IOS Optimized Edge Routing Command Reference*.

Step 4	**`mode select-exit`** `{`**`best`** `|` **`good`**`}`
	Example:
	`Router(config-oer-mc)# mode select-exit best`
		Configures exit selection settings.
		• Use the **select-exit** keyword to configure the master controller to select either the best available exit when the **best** keyword is entered or the first in-policy exit when the **good** keyword is entered.
		• In this example, OER will select the best available exit.
		Note Only the syntax relevant to this task is displayed. For more details, see the *Cisco IOS Optimized Edge Routing Command Reference*.

	Command or Action	Purpose
Step 5	**resolve range priority** *value* **Example:** `Router(config-oer-mc)# resolve range priority 1`	Sets policy priority or resolves policy conflicts. • This command is used to set the priorities when multiple policies are configured for the same prefix. When this command is configured, the policy with the highest priority will be selected to determine the policy decision. • The **priority** keyword is used to specify the priority value. Setting the number 1 assigns the highest priority to a policy. Setting the number 10 assigns the lowest priority. • Each policy must be assigned a different priority number. • In this example, the priority for range policies is set to 1. **Note** Only the syntax relevant to this task is displayed. For more details, see the *Cisco IOS Optimized Edge Routing Command Reference*.
Step 6	**resolve utilization priority** *value* **variance** *percentage* **Example:** `Router(config-oer-mc)# resolve utilization priority 2 variance 20`	Sets policy priority or resolves policy conflicts. • This command is used to set the priorities when multiple policies are configured for the same prefix. When this command is configured, the policy with the highest priority will be selected to determine the policy decision. • The **priority** keyword is used to specify the priority value. Setting the number 1 assigns the highest priority to a policy. Setting the number 10 assigns the lowest priority. • Each policy must be assigned a different priority number. • The **variance** keyword is used to set an allowable variance for a user-defined policy. This keyword configures the allowable percentage that an exit link or prefix can vary from the user-defined policy value and still be considered equivalent. • In this example, the priority for utilization policies is set to 2 with a 20 percent variance. **Note** Only the syntax relevant to this task is displayed. For more details, see the *Cisco IOS Optimized Edge Routing Command Reference*.
Step 7	**no resolve delay** **Example:** `Router(config-oer-mc)# no resolve delay`	Disables any priority for delay performance policies. **Note** Only the syntax relevant to this task is displayed. For more details, see the *Cisco IOS Optimized Edge Routing Command Reference*.

	Command or Action	Purpose
Step 8	**no resolve loss** **Example:** `Router(config-oer-mc)# no resolve loss`	Disables any priority for loss performance policies. **Note** Only the syntax relevant to this task is displayed. For more details, see the *Cisco IOS Optimized Edge Routing Command Reference*.
Step 9	**max range receive percent** *percentage* **Example:** `Router(config-oer-mc)# max range receive percent 20`	Specifies the upper limit of the inbound (receive) traffic utilization range between all the entrance links on the border routers. • The **percent** keyword and *percentage* argument are used to specify the range percentage. • In this example, the inbound traffic utilization range between all the entrance links on the border routers must be within 20 percent.
Step 10	**border** *ip-address* [**key-chain** *key-chain-name*] **Example:** `Router(config-oer-mc)# border 10.1.1.2 key-chain border1_OER`	Enters OER-managed border router configuration mode to establish communication with a border router. • An IP address is configured to identify the border router. • At least one border router must be specified to create an OER-managed network. A maximum of ten border routers can be controlled by a single master controller. **Note** The **key-chain** keyword and *key-chain-name* argument must be entered when a border router is initially configured. However, this keyword is optional when reconfiguring an existing border router.
Step 11	**interface** *type number* **external** **Example:** `Router(config-oer-mc-br)# interface Ethernet 1/0 external`	Configures a border router interface as an OER-managed external interface. • External interfaces are used to forward traffic and for active monitoring. • A minimum of two external border router interfaces are required in an OER-managed network. At least one external interface must be configured on each border router. A maximum of 20 external interfaces can be controlled by single master controller. **Tip** Configuring an interface as an OER-managed external interface on a router enters OER border exit interface configuration mode. In this mode, you can configure maximum link utilization or cost-based optimization for the interface. **Note** Entering the **interface** command without the **external** or **internal** keyword places the router in global configuration mode and not OER border exit configuration mode. The **no** form of this command should be applied carefully so that active interfaces are not removed from the router configuration.

	Command or Action	Purpose

Step 12	**maximum utilization receive** {**absolute** *kbps* | **percent** *percentage*}
	Example:
	`Router(config-oer-mc-br-if)# maximum utilization receive percent 90`
		Sets the maximum inbound (receive) traffic utilization for the configured OER-managed link interface.
		• Use the **absolute** keyword and *kbps* argument to specify the absolute value, in kilobytes per second (kbps), of the throughput for all the entrance links.
		• Use the **percent** keyword and *percentage* argument to specify the maximum utilization as a percentage of bandwidth received by all the entrance links.
		• In this example, the maximum utilization of inbound traffic on this entrance link on the border router must be 90 percent, or less.

Step 13	**downgrade bgp community** *community-number* **Example:** `Router(config-oer-mc-br-if)# downgrade bgp community 4:5`	Specifies downgrade options for BGP advertisement for the configured OER-managed entrance link interface. • Use the **community** keyword and *community-number* argument to specify a BGP community number that will be added to the BGP advertisement. • In this example, the BGP community number 4:5 will be added to BGP advertisements to packets sent from this entrance link if it is not selected as the best entrance link.
Step 14	**exit** **Example:** `Router(config-oer-mc-br-if)# exit`	Exits OER border exit interface configuration mode and returns to OER-managed border router configuration mode.
Step 15	Repeat Step 14 twice to return to global configuration mode.	—
Step 16	**oer-map** *map-name sequence-number* **Example:** `Router(config)# oer-map INSIDE_LEARN 10`	Enters OER map configuration mode to configure an OER map to apply policies to selected IP prefixes. • *Only one match clause can be configured for each OER map sequence.* • Deny sequences are first defined in an IP prefix list and then applied with a **match** command. • The example creates an OER map named INSIDE_LEARN.

Step 17	**match oer learn** {**delay** | **inside** | **throughput**}
	Example:
	`Router(config-oer-map)# match oer learn inside`
		Creates a match clause entry in an OER map to match OER learned prefixes.
		• Prefixes can be configured to learn prefixes that are inside prefixes or prefixes based on lowest delay, or highest outbound throughput.
		• Only a single match clause can be configured for each OER map sequence.
		• The example creates a match clause entry that matches traffic classes learned using inside prefixes.

	Command or Action	Purpose

Step 18	`set delay {relative percentage | threshold maximum}`
	Example:
	`Router(config-oer-map)# set delay threshold 200`
		Creates a set clause entry to configure the delay threshold.
		• The delay threshold can be configured as a relative percentage or as an absolute value for match criteria.
		• The **relative** keyword is used to configure a relative delay percentage. The relative delay percentage is based on a comparison of short-term and long-term measurements.
		• The **threshold** keyword is used to configure the absolute maximum delay period in milliseconds.
		• The example creates a set clause that sets the absolute maximum delay threshold to 200 milliseconds for traffic that is matched in the same OER map sequence.

Step 19	`set mode route control`
	Example:
	`Router(config-oer-map)# set mode route control`
		Creates a set clause entry to configure route control for matched traffic.
		• In control mode, the master controller analyzes monitored traffic classes and implements changes based on policy parameters.
		• In this example, a set clause that enables OER control mode is created.
		Note Only the syntax applicable to this task is shown. For more details, see the *Cisco IOS Optimized Edge Routing Command Reference*.

Step 20	`end`
	Example:
	`Router(config-oer-map)# end`
		(Optional) Exits OER map configuration mode and returns to privileged EXEC mode.

Manually Verifying the OER Route Control Changes

OER automatically verifies route control changes in the network using NetFlow output. OER monitors the NetFlow messages and uncontrols a traffic class if a message does not appear to verify the route control change. Perform the steps in this optional task if you want to manually verify that the traffic control implemented by the OER control phase actually changes the traffic flow, and brings the OOP event to be in-policy. All the steps are optional and are not in any order. The information from these steps can verify that a specific prefix associated with a traffic class has been moved to another exit or entrance link interface, or that it is being controlled by OER. The first three commands are entered at the master controller, the last command is entered at a border router. For more details about other OER show commands, see the *Cisco IOS Optimized Edge Routing Command Reference*.

SUMMARY STEPS

1. **enable**
2. **show logging** [**slot** *slot-number* | **summary**]
3. **show oer master prefix** *prefix* [**detail**]
4. Move to a border router to enter the next step.

5. **enable**
6. **show oer border routes** {**bgp** | **static**}

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

```
Router> enable
```

Step 2 **show logging** [**slot** *slot-number* | **summary**]

This command is used to display the state of system logging (syslog) and the contents of the standard system logging buffer. Using optional delimiters, this example shows the logging buffer with OER messages for the prefix 10.1.1.0 that is OOP and has a route change.

```
Router# show logging | i 10.1.1.0

*Apr 26 22:58:20.919: %OER_MC-5-NOTICE: Discovered Exit for prefix 10.1.1.0/24, BR
10.10.10.1, i/f Et9/0
*Apr 26 23:03:14.987: %OER_MC-5-NOTICE: Route changed 10.1.1.0/24, BR 10.10.10.1, i/f
Se12/0, Reason Delay, OOP Reason Timer Expired
*Apr 26 23:09:18.911: %OER_MC-5-NOTICE: Passive REL Loss OOP 10.1.1.0/24, loss 133, BR
10.10.10.1, i/f Se12/0, relative loss 23, prev BR Unknown i/f Unknown
*Apr 26 23:10:51.123: %OER_MC-5-NOTICE: Route changed 10.1.1.0/24, BR 10.10.10.1, i/f
Et9/0, Reason Delay, OOP Reason Loss
```

Step 3 **show oer master prefix** *prefix* [**detail**]

This command is used to display the status of monitored prefixes. The output from this command includes information about the source border router, current exit interface, prefix delay, and egress and ingress interface bandwidth. In this example, the output is filtered for the prefix 10.1.1.0 and shows that the prefix is currently in a holddown state. Only syntax relevant to this task, is shown in this step.

```
Router# show oer master prefix 10.1.1.0

Prefix         State   Time Curr BR     CurrI/F     Protocol
           PasSDly PasLDly  PasSUn  PasLUn PasSLos PasLLos
           ActSDly ActLDly  ActSUn  ActLUn   EBw   IBw
--------------------------------------------------------------------------------
10.1.1.0/24   HOLDDOWN  42 10.10.10.1   Et9/0      STATIC
                16      16       0       0       0       0
                 U       U       0       0      55       2
```

Step 4 Move to a border router to enter the next step.

The next command is entered on a border router, not the master controller.

Step 5 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

```
Router> enable
```

Step 6 **show oer border routes** {**bgp** | **static**}

This command is entered on a border router. This command is used to display information about OER controlled routes on a border router. You can display information about BGP or static routes. In this example, the output shows that prefix 10.1.1.0 is being controlled by OER.

```
Router# show oer border routes bgp
```

```
OER BR 10.10.10.1 ACTIVE, MC 10.10.10.3 UP/DOWN: UP 00:10:08,
   Auth Failures: 0
   Conn Status: SUCCESS, PORT: 3949
BGP table version is 12, local router ID is 10.10.10.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
OER Flags: C - Controlled, X - Excluded, E - Exact, N - Non-exact, I - Injected

    Network          Next Hop        OER    LocPrf Weight Path
*> 10.1.1.0/24     10.40.40.2         CE         0 400 600 i
```

Configuring Traceroute Reporting

Perform this task at the master controller to configure traceroute reporting. When using an OER active probe there are situations when a host address does not respond to the OER probe message. The reason for no response to the probe message may be due to a firewall or other network issue but OER assumes the host address to be unreachable and releases control of the prefix. Prior to traceroute reporting there was no method for measuring the delay per hop for situations such as an unexpected round trip delay value being reported for a traffic class on an exit link. The solution for both the non-responding target address and the lack of per-hop delay information involves using UDP, and optionally TCP, traceroutes. Traceroute reporting is configured on a master controller, but the traceroute probes are sourced from the border router exits.

In this task, the three methods of configuring traceroute probes are used. Periodic and policy-based traceroute reporting are configured with the **set traceroute reporting** command using an OER map. On-demand traceroute probes are triggered by entering the **show oer master prefix** command with certain parameters. This task also shows to modify the time interval between traceroute probes using the **traceroute probe-delay** command.

When traceroute reporting is enabled, the default time interval between traceroute probes is 1000 milliseconds.

Prerequisites

This task requires the master controller and border routers to be running Cisco IOS Release 12.3(14)T, 12.2(33)SRB, or later releases.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **oer master**
4. **traceroute probe-delay** *milliseconds*
5. **exit**
6. **oer-map** *map-name sequence-number*
7. **match oer learn** {**delay** | **throughput**}
8. **set traceroute reporting** [**policy** {**delay** | **loss** | **unreachable**}]

9. **end**
10. **show oer master prefix** [**detail** | **learned** [**delay** | **throughput**] | *prefix* [**detail** | **policy** | **traceroute** [*exit-id* | *border-address* | **current**] [**now**]]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**oer master** **Example:** `Router(config)# oer master`	Enters OER master controller configuration mode to configure a router as a master controller and to configure global operations and policies.
Step 4	**traceroute probe-delay** *milliseconds* **Example:** `Router(config-oer-mc)# traceroute probe-delay 500`	Sets the time interval between traceroute probe cycles. • The default time interval between traceroute probes is 1000 milliseconds. • The example sets the probe interval to a 500 milliseconds.
Step 5	**exit** **Example:** `Router(config-oer-mc)# exit`	Exits OER master controller configuration mode, and returns to global configuration mode.
Step 6	**oer-map** *map-name sequence-number* **Example:** `Router(config)# oer-map TRACEROUTE 10`	Enters oer-map configuration mode to configure an OER map to apply policies to selected IP prefixes. • *Only one match clause can be configured for each oer-map sequence.* • The example creates an OER map named TRACEROUTE.

Step 7	**match oer learn** {**delay** | **throughput**}
	Example:
	`Router(config-oer-map)# match oer learn delay`
		Creates a match clause entry in an oer-map to match learned prefixes.
		• Can be configured to learn prefixes based on highest delay or highest outbound throughput.
		• Only a single match clause can be configured for each oer-map sequence.
		• The example creates a match clause entry that matches traffic learned based on highest delay.

	Command or Action	Purpose
Step 8	**set traceroute reporting** [**policy** {**delay** | **loss** | **unreachable**}] **Example:** `Router(config-oer-map)# set traceroute reporting`	Configures an OER map to enable traceroute reporting. • Monitored prefixes must be included in an OER map. These can be learned or manually selected prefixes. • Entering this command with no keywords enables continuous monitoring. • Entering this command the policy keyword enables policy-based trace route reporting.

Step 9	**end**
	Example:
	`Router(config-oer-map)# end`
		Exits OER master controller configuration mode, and returns to privileged EXEC mode.

Step 10	**show oer master prefix** [**detail** | **learned** [**delay** | **throughput**] | *prefix* [**detail** | **policy** | **traceroute** [*exit-id* | *border-address* | **current**] [**now**]]] **Example:** `Router# show oer master prefix 10.5.5.5 traceroute now`	Displays the status of monitored prefixes. • An on-demand traceroute probe is initiated by entering the **current** and **now** keywords. • The **current** keyword displays the results of the most recent traceroute probe for the current exit. • Traceroute probe results can be displayed for the specified border router exit by entering the **exit-id** or **border-address** keyword. • The example initiates an on-demand traceroute probe for the 10.5.5.55 prefix.

Configuration Examples for Using OER to Control Traffic Classes and Verify the Route Control Changes

The configuration examples in this section show how to configure OER to control traffic classes, verify the network performance, and configure troubleshooting.

- Enabling OER Route Control Mode: Example, page 26
- Setting a Tag Value for Injected OER Static Routes: Example, page 27
- Setting a BGP Local Preference Value for OER Controlled BGP Routes: Example, page 27
- Controlling Application Traffic: Example, page 27
- Controlling Voice Application Traffic: Example, page 28
- Enforcing Entrance Link Selection with Load Balancing for an Inside Prefix: Example, page 28
- Manually Verifying the OER Route Control Changes: Examples, page 29
- Configuring Traceroute Reporting: Examples, page 29

Enabling OER Route Control Mode: Example

The following example shows how to configure the master controller to use route control mode:

```
Router(config)# oer master
Router(config-oer-mc)# mode route control
Router(config-oer-mc)# end
```

Setting a Tag Value for Injected OER Static Routes: Example

The following example shows how to set a tag value for an injected static route to allow the routes to be uniquely identified. A static route may be injected by OER to control the traffic defined by a traffic class when it goes out-of-policy. By default, OER uses a tag value of 5000 for injected static routes. In this task, the OER route control mode is configured globally with the **mode** command in OER master controller configuration mode and any injected static routes will be tagged with a value of 15000.

```
Router(config)# oer master
Router(config-oer-mc)# mode route control
Router(config-oer-mc)# mode metric static tag 15000
Router(config-oer-mc)# end
```

Setting a BGP Local Preference Value for OER Controlled BGP Routes: Example

The following example shows how to set a BGP local preference attribute value. OER uses the BGP Local_Pref value to influence the BGP best path selection on internal BGP (iBGP) neighbors as a method of enforcing exit link selection. By default, OER uses a Local_Pref value of 5000. In this task, route control is enabled for traffic matching a prefix list and the BGP local preference value of 60000 is set.

```
Router(config)# oer-map BLUE 10
Router(config-oer-map)# match ip address prefix-list BLUE
Router(config-oer-map)# set mode route control
Router(config-oer-map)# set mode metric bgp local-pref 60000
Router(config-oer-map)# end
```

Controlling Application Traffic: Example

The following example shows how to use policy-based routing (PBR) to allow OER to control specified application traffic classes. Application traffic such as Telnet traffic is delay sensitive. Long TCP delays can make Telnet sessions difficult to use. This example is configured on a master controller and matches Telnet traffic sourced from the 192.168.1.0/24 network and applies a policy to ensure it is sent out through exit links with that have a response time that is equal to or less than 30 milliseconds:

```
Router(config)# ip access-list extended TELNET
Router(config-ext-nacl)# permit tcp 192.168.1.0 0.0.0.255 any eq telnet
Router(config-ext-nacl)# exit
Router(config)# oer-map SENSITIVE
Router(config-route-map)# match ip address access-list TELNET
Router(config-route-map)# set mode route control
Router(config-route-map)# set delay threshold 30
Router(config-route-map)# set resolve delay priority 1 variance 20
Router(config-route-map)# end
```

The following example shows TCP application traffic filtered based on port 23 (Telnet):

```
Router# show oer master appl tcp 23 23 dst policy
Prefix              Appl Prot       Port                 Port Type       Policy
--------------------------------------------------------------------------------
10.1.1.0/24        tcp             [23, 23]             src             10
```

Controlling Voice Application Traffic: Example

The following example shows how to control application traffic such as voice traffic. An OER map for voice traffic is configured using a traffic class that represents voice traffic with a DSCP value of ef. Route control is enabled. The **policy-rules** command applies the configuration from the OER map named VOICE_MAP to the master controller configuration and overwrites any previous OER map configuration. To run this task, both the master controller and border routers must be running Cisco IOS Release 12.4(9)T, 12.2(33)SRB, or later release.

```
Router> enable
Router# configure terminal
Router(config)# ip prefix-list CONFIG_TRAFFIC_CLASS seq 10 permit 10.1.5.0/24
Router(config)# ip access-list extended VOICE_TRAFFIC_CLASS
Router(config-ext-nacl)# permit udp any range 16384 32767 10.1.5.0 0.0.0.15 range 16384
32767 dscp ef
Router(config-ext-nacl)# exit
Router(config)# oer-map VOICE_MAP 10
Router(config-oer-map)# match ip address access-list VOICE_TRAFFIC_CLASS
Router(config-oer-map)# set active-probe jitter 10.1.5.1 target-port 2000 codec g729a
Router(config-oer-map)# set delay threshold 1000
Router(config-oer-map)# set loss relative 25
Router(config-oer-map)# set probe-frequency 20
Router(config-oer-map)# set jitter threshold 30
Router(config-oer-map)# set mos threshold 4.0 percent 25
Router(config-oer-map)# exit
Router(config)# oer master
Router(config-oer-mc)# mode route control
Router(config-oer-mc)# policy-rules VOICE_MAP
Router(config-oer-mc)# end
```

Enforcing Entrance Link Selection with Load Balancing for an Inside Prefix: Example

The following example shows how to enforce an entrance link selection for learned inside prefixes using the BGP autonomous system number community prepend technique:

```
Router> enable
Router# configure terminal
Router(config)# oer master
Router(config-oer-mc)# mode select-exit best
Router(config-oer-mc)# resolve range priority 1
Router(config-oer-mc)# resolve utilization priority 2 variance 20
Router(config-oer-mc)# no resolve delay
Router(config-oer-mc)# no resolve loss
Router(config-oer-mc)# max range receive percent 35
Router(config-oer-mc)# border 10.1.1.2 key-chain oer
Router(config-oer-mc-br)# interface ethernet1/0 external
Router(config-oer-mc-br-if)# maximum utilization receive absolute 2500
Router(config-oer-mc-br-if)# downgrade bgp community 3:1
Router(config-oer-mc-br-if)# exit
Router(config-oer-mc-br)# exit
Router(config-oer-mc)# exit
Router(config)# oer-map INSIDE_LEARN 10
Router(config-oer-map)# match oer learn inside
Router(config-oer-map)# set delay threshold 400
Router(config-oer-map)# set mode route control
Router(config-oer-map)# end
```

Manually Verifying the OER Route Control Changes: Examples

The following examples show how to manually verify that the traffic control implemented by the OER control phase actually changes the traffic flow and brings the OOP event to be in-policy. On the master controller the **show logging** command is used to display the state of system logging (syslog) and the contents of the standard system logging buffer. Using optional delimiters, the logging buffer can be displayed with OER messages for a specific prefix. The **show oer master prefix** command displays the status of monitored prefixes. On the border router, the **show oer border routes** command displays information about OER controlled BGP or static routes on the border router. For example output of these commands, see the "Manually Verifying the OER Route Control Changes" section on page 22.

Master Controller

```
Router# show logging | i 10.1.1.0
Router# show oer master prefix 10.1.1.0
Router# end
```

Border Router

```
Router# show oer border routes static
Router# show oer border routes bgp
Router# end
```

Configuring Traceroute Reporting: Examples

The following example, starting in global configuration mode, configures continuous traceroute reporting for traffic classes learned on the basis of delay:

```
Router(config)# oer master
Router(config-oer-mc)# traceroute probe-delay 10000
Router(config-oer-mc)# exit
Router(config)# oer-map TRACE 10
Router(config-oer-map)# match oer learn delay
Router(config-oer-map)# set traceroute reporting
Router(config-oer-map)# end
```

The following example, starting in privileged EXEC mode, initiates an on-demand traceroute probe for the 10.5.5.5 prefix:

```
Router# show oer master prefix 10.5.5.55 traceroute current now
Path for Prefix: 10.5.5.0/24         Target: 10.5.5.5
Exit ID: 2, Border: 10.1.1.3         External Interface: Et1/0
Status: DONE, How Recent: 00:00:08 minutes old
Hop  Host            Time(ms) BGP
1    10.1.4.2        8        0
2    10.1.3.2        8        300
3    10.5.5.5        20       50
```

Where to Go Next

This module described the OER control and verify phases and it has assumed that you started with the Cisco IOS Optimized Edge Routing Overview module, followed by the Setting Up OER Network Components module. The control and verify phases are the last two phases in the OER performance loop. To learn more about the other OER phases, read through the other modules in the following list:

- Using OER to Profile the Traffic Classes
- Measuring the Traffic Class Performance and Link Utilization Using OER
- Configuring and Applying OER Policies
- Using OER to Control Traffic Classes and Verify the Route Control Changes

After you understand the various OER phases, you may want to review the OER Solutions modules that are listed under "Related Documents" section on page 30.

Additional References

The following sections provide references related to using OER to control traffic classes and verify the route control changes.

Related Documents

Related Topic	Document Title
Cisco OER technology overview	"Cisco IOS Optimized Edge Routing Overview" module
Concepts and configuration tasks required to set up OER network components.	"Setting Up OER Network Components" module
OER solution module: voice traffic optimization using OER active probes.	"OER Voice Traffic Optimization Using Active Probes" module
OER solution module: configuring VPN IPsec/GRE tunnel interfaces as OER-managed exit links.	"Configuring VPN IPsec/GRE Tunnel Interfaces As OER-Managed Exit Links" module
Cisco OER commands: complete command syntax, command mode, command history, defaults, usage guidelines and examples	*Cisco IOS Optimized Edge Routing Command Reference*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Using OER to Control Traffic Classes and Verify the Route Control Changes

Table 1 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.3(8)T, 12.2(33)SRB, or a later release appear in the table.

For information on a feature in this technology that is not documented here, see the "Cisco IOS Optimized Edge Routing Feature Roadmap."

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Using OER to Control Traffic Classes and Verify the Route Control Changes

Feature Name	Releases	Feature Configuration Information
Optimized Edge Routing	12.3(8)T 12.2(33)SRB	OER was introduced.
OER Support for Cost-Based Optimization and Traceroute Reporting	12.3(14)T 12.2(33)SRB	The OER Support for Traceroute Reporting feature allows you to monitor prefix performance on a hop-by-hop basis. Delay, loss, and reachability measurements are gathered for each hop from the probe source (border router) to the target prefix. The following sections provide information about this feature: • OER Troubleshooting Using Traceroute Reporting, page 8 • Configuring Traceroute Reporting, page 24 • Manually Verifying the OER Route Control Changes: Examples, page 29 The following commands were introduced or modified by this feature: **set traceroute reporting**, **traceroute probe-delay**, and **show oer master prefix**.

Table 1 Feature Information for Using OER to Control Traffic Classes and Verify the Route Control Changes

Feature Name	Releases	Feature Configuration Information
OER Application-Aware Routing: PBR	12.4(2)T 12.2(33)SRB	The OER Application-Aware Routing: PBR feature introduces the capability to optimize IP traffic based on the type of application that is carried by the monitored prefix. Independent policy configuration is applied to the subset (application) of traffic. The following sections provide information about this feature: • Policy Route, page 6 • Controlling Application Traffic, page 13 • Controlling Application Traffic: Example, page 27 The following commands were introduced or modified by this feature: **debug oer border pbr**, **debug oer master prefix**, **match ip address (OER)**, **show oer master active-probes**, and **show oer master appl**.
OER Voice Traffic Optimization	12.4(6)T 12.2(33)SRB	The OER Voice Traffic Optimization feature introduced support for outbound optimization of voice traffic based on the voice metrics, jitter and Mean Opinion Score (MOS). Jitter and MOS are important quantitative quality metrics for voice traffic and these voice metrics are measured using OER active probes. The following sections provide information about this feature: • Controlling Voice Application Traffic: Example, page 28 The following commands were introduced or modified by this feature: **active-probe**, **jitter**, **mos**, **resolve**, **set jitter**, **set mos**, **set probe**, **set resolve**, **show oer master active-probes**, **show oer master policy**, and **show oer master prefix**.

Table 1 Feature Information for Using OER to Control Traffic Classes and Verify the Route Control Changes

Feature Name	Releases	Feature Configuration Information
OER BGP Inbound Optimization	12.4(9)T 12.2(33)SRB	OER BGP inbound optimization supports best entrance selection for traffic that originates from prefixes outside an autonomous system destined for prefixes inside the autonomous system. External BGP (eBGP) advertisements from an autonomous system to an Internet service provider (ISP) can influence the entrance path for traffic entering the network. OER uses eBGP advertisements to manipulate the best entrance selection. The following sections provide information about this feature: • OER Traffic Class Control Techniques, page 4 • OER Entrance Link Selection Control Techniques, page 7 • Enforcing Entrance Link Selection with Load Balancing for an Inside Prefix, page 16 • Enforcing Entrance Link Selection with Load Balancing for an Inside Prefix: Example, page 28 The following commands were introduced or modified by this feature: **clear oer master prefix**, **downgrade bgp**, **inside bgp**, **match ip address (OER)**, **match oer learn**, **max range receive**, **maximum utilization receive**, **show oer master prefix**.
OER DSCP Monitoring	12.4(9)T 12.2(33)SRB	OER DSCP Monitoring introduced automatic learning of traffic classes based on protocol, port numbers, and DSCP value. Traffic classes can be defined by a combination of keys comprising of protocol, port numbers, and DSCP values, with the ability to filter out traffic that is not required, and the ability to aggregate the traffic in which you are interested. Information such as protocol, port number, and DSCP information is now sent to the master controller database in addition to the prefix information. The new functionality allows OER to both actively and passively monitor application traffic. The following sections provide information about this feature: • OER Traffic Class Control Techniques, page 4 • Controlling Application Traffic, page 13 The following commands were introduced or modified by this feature: **show oer border passive applications**, **show oer border passive cache**, **show oer border passive learn**, **show oer master appl**, **traffic-class aggregation**, **traffic-class filter**, and **traffic-class keys**.

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

OER Voice Traffic Optimization Using Active Probes

First Published: August 14, 2006
Last Updated: February 28, 2007

This module documents an Optimized Edge Routing (OER) solution that supports outbound optimization of voice traffic based on the voice metrics, jitter and Mean Opinion Score (MOS). Jitter and MOS are important quantitative quality metrics for voice traffic and these voice metrics are measured using Optimized Edge Routing (OER) active probes.

OER provides automatic route optimization and load distribution for multiple connections between networks. OER is an integrated Cisco IOS solution that allows you to monitor IP traffic flows and then define policies and rules based on prefix performance, link load distribution, link bandwidth monetary cost, and traffic type. OER provides active and passive monitoring systems, dynamic failure detection, and automatic path correction. Deploying OER enables intelligent load distribution and optimal route selection in an enterprise network.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for OER Voice Traffic Optimization Using Active Probes" section on page 19.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for OER Voice Traffic Optimization Using Active Probes

Before implementing OER optimization for voice traffic, you need to understand an overview of how OER works and how to set up OER network components. See the "Cisco IOS Optimized Edge Routing Overview" and "Setting Up OER Network Components" modules for more details. For a list of other OER configuration modules, see the "Where to Go Next" section on page 18 and the "Related Documents" section on page 18.

Information About OER Voice Traffic Optimization Using Active Probes

Before you configure OER voice traffic optimization, you should understand the following concepts:

Voice Quality on IP Networks

Voice packets traveling through an IP network are no different from data packets. In the plain old telephone system (POTS), voice traffic travels over circuit-switched networks with predetermined paths and each phone call is given a dedicated connection for the duration of the call. Voice traffic using POTS has no resource contention issues, but voice traffic over an IP network has to contend with factors such as delay, jitter, and packet loss, which can affect the quality of the phone call.

Delay

Delay (also referred as latency) for voice packets is defined as the delay between when the packet was sent from the source device and when it arrived at a destination device. Delay can be measured as one-way delay or round-trip delay. The largest contributor to latency is caused by network transmission delay. Round-trip delay affects the dynamics of conversation and is used in Mean Opinion Score (MOS) calculations. One-way delay is used for diagnosing network problems. A caller may notice a delay of 200 milliseconds and try to speak just as the other person is replying because of packet delay. The telephone industry standard specified in ITU-T G.114 recommends the maximum desired one-way delay be no more than 150 milliseconds. Beyond a one-way delay of 150 milliseconds, voice quality is affected. With a round-trip delay of 300 milliseconds or more, users may experience annoying talk-over effects.

Jitter

Jitter means interpacket delay variance. When multiple packets are sent consecutively from source to destination, for example, 10 ms apart, and if the network is behaving ideally, the destination should be receiving them 10 ms apart. But if there are delays in the network (like queuing, arriving through alternate routes, and so on) the arrival delay between packets might be greater than or less than 10 ms. Using this example, a positive jitter value indicates that the packets arrived more than 10 ms apart. If the packets arrive 12 ms apart, then positive jitter is 2 ms; if the packets arrive 8 ms apart, then negative jitter is 2 ms. For delay-sensitive networks like VoIP, positive jitter values are undesirable, and a jitter value of 0 is ideal.

Packet Loss

Packet loss can occur due an interface failing, a packet being routed to the wrong destination, or congestion in the network. Packet loss for voice traffic leads to the degradation of service in which a caller hears the voice sound with breaks. Although average packet loss is low, voice quality may be affected by a short series of lost packets.

Mean Opinion Score (MOS)

With all the factors affecting voice quality, many people ask how voice quality can be measured. Standards bodies like the ITU have derived two important recommendations: P.800 (MOS) and P.861 (Perceptual Speech Quality Measurement [PSQM]). P.800 is concerned with defining a method to derive a Mean Opinion Score of voice quality. MOS scores range between 1 representing the worst voice quality, and 5 representing the best voice quality. A MOS of 4 is considered “toll-quality” voice.

Probes Used by OER

OER uses some of the IP SLA probes to help gather the data OER requires to make its decisions.

Cisco IOS IP SLAs

Cisco IOS IP SLAs are an embedded feature set in Cisco IOS software and they allow you to analyze IP service levels for IP applications and services, to increase productivity, to lower operational costs, and to reduce occurrences of network congestion or outages. IP SLAs use active traffic monitoring—the generation of traffic in a continuous, reliable, and predictable manner—for measuring network performance. The accuracy of measured data is enhanced by enabling the IP SLAs Responder, available in Cisco routers, on the destination device. For more details about IP SLAs, see the *Cisco IOS IP SLAs Configuration Guide*.

Active Probe Types Used by OER

The following types of active probes can be configured:

ICMP Echo—A ping is sent to the target address. OER uses ICMP Echo probes, by default, when an active probe is automatically generated. Configuring an ICMP echo probe does not require knowledgeable cooperation from the target device. However, repeated probing could trigger an Intrusion Detection System (IDS) alarm in the target network. If an IDS is configured in a target network that is not under your control, we recommend that you notify the administrator of this target network.

Jitter—A jitter probe is sent to the target address. A target port number must be specified. A remote responder must be enabled on the target device, regardless of the configured port number.

TCP Connection—A TCP connection probe is sent to the target address. A target port number must be specified. A remote responder must be enabled if TCP messages are configured to use a port number other than TCP port number 23, which is well-known.

UDP Echo—A UDP echo probe is sent to the target address. A target port number must be specified. A remote responder must be enabled on the target device, regardless of which port number is configured.

Probe Frequency

In Cisco IOS Release 12.4(4)T and earlier releases, the frequency of an active probe used by OER was set to 60 seconds. In Cisco IOS Release 12.4(6)T and 12.2(33)SRB, the frequency can be increased for each policy by configuring a lower time-interval between two probes. Increased probe frequency can reduce the response time and provide a better approximation of the MOS-low count percentage

OER Voice Traffic Optimization Using Active Probes

OER voice traffic optimization provides support for outbound optimization of voice traffic on the basis of the voice performance metrics, delay, packet loss, jitter, and MOS. Delay, packet loss, jitter and MOS are important quantitative quality metrics for voice traffic, and these voice metrics are measured using OER active probes. In Cisco IOS Release 12.4(4)T and earlier releases, OER probes could measure delay and packet loss, but not jitter and MOS. The IP SLA jitter probe is integrated with OER to measure jitter (source to destination) and the MOS score in addition to measuring delay and packet loss. The jitter probe requires a responder on the remote side just like the UDP Echo probe. Integration of the IP SLA jitter probe type in OER enhances the ability of OER to optimize voice traffic. OER policies can be configured to set the threshold and priority values for the voice performance metrics: delay, packet loss, jitter, and MOS.

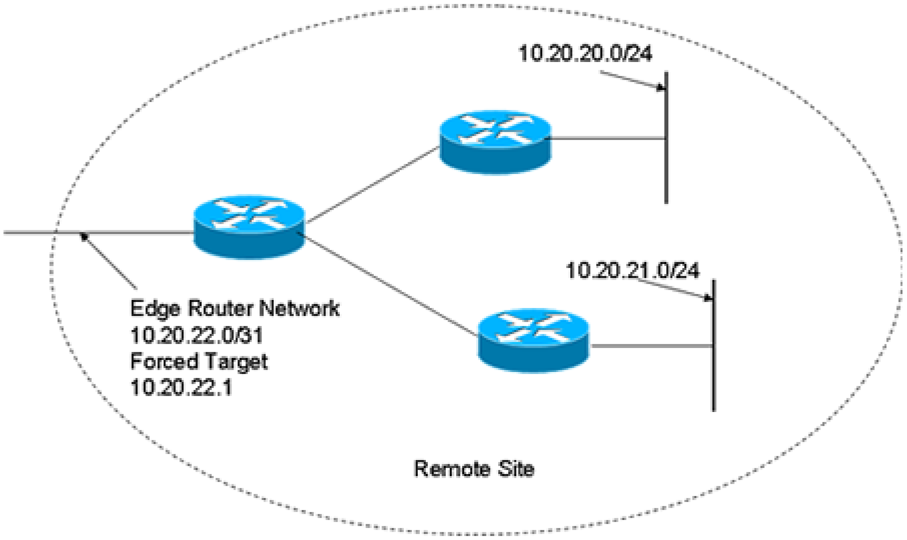
Configuring an OER policy to measure jitter involves configuring only the threshold value and not relative changes (used by other OER features) because for voice traffic, relative jitter changes have no meaning. For example, jitter changes from 5 milliseconds to 25 milliseconds are just as bad in terms of voice quality as jitter changes from 15 milliseconds to 25 milliseconds. If the short-term average (measuring the last 5 minutes) jitter is higher than the jitter threshold, the prefix is considered out-of-policy due to jitter. OER then probes all exits, and the exit with the least jitter is selected as the best exit.

MOS policy works in a different way. There is no meaning to average MOS values, but there is meaning to the number of times that the MOS value is below the MOS threshold. For example, if the MOS threshold is set to 3.85 and if 3 out of 10 MOS measurements are below the 3.85 MOS threshold, the MOS-low-count is 30 percent. When OER runs a policy configured to measure MOS, both the MOS threshold value and the MOS-low-count percentage are considered. A prefix is considered out-of-policy if the short term (during the last 5 minutes) MOS-low-count percentage is greater than the configured value for a given MOS threshold. OER then probes all exits, and the exit with the highest MOS value is selected as the best exit.

OER Forced Target Assignment

In Cisco IOS Release 12.4(4)T and earlier releases, the OER active probe target is assigned to the longest matched prefix. There are some scenarios where you may want to use a target that does not match the destination prefix. The example in Figure 1 explains a scenario in which configuring an OER forced target assignment is more appropriate than using the longest match prefix.

Figure 1 OER Forced Target Assignment Scenario



In Figure 1 we want to probe IP address 10.20.22.1 (at the edge of the network) for either network 10.20.21.0/24 or 10.20.22.0/24. Jitter is less likely to be introduced within the network so probing the edge of the network gives a measurement that is close to probing the final destination.

Forced target assignment allows you to assign a target to a group of prefixes or an application, even if they are not the longest match prefixes. Assigning a target can determine the true delay to the edge of a network rather than delay to an end host.

How to Configure OER Voice Traffic Optimization Using Active Probes

Configuring OER to optimize voice traffic using active probes involves several decisions and subsequent branching tasks. The first step is to identify the traffic to be optimized and decide whether to use a prefix list or an access list. Use a prefix list to identify all traffic, including voice traffic, with a specific set of destination prefixes. Use an access list to identify only voice traffic with a specific destination prefix and carried over a specific protocol.

The second step in optimizing voice traffic is to configure active probing using the **active-probe** or **set active-probe** command to specify the type of active probe to be used. In Cisco IOS Release 12.4(6)T and 12.2(33)SRB, the ability to set a forced target assignment for the active probe was introduced.

The final step in optimizing voice traffic is to configure an OER policy to set the performance metrics that you want OER to apply to the identified traffic.

Perform one of the first two optional tasks, depending on whether you want to use a prefix list or an access list to identify the traffic to be optimized. The third task can be used with traffic identified using an access list, and it also demonstrates how to use a forced target assignment. For an example configuration that can be used with traffic identified using a prefix list, see the "Optimizing Traffic (Including Voice Traffic) Using Active Probes: Example" section on page 16.

- Identifying Traffic for OER Using a Prefix List, page 6
- Identifying Voice Traffic to Optimize Using an Access List, page 7
- Configuring OER Voice Probes with a Target Assignment, page 8

Identifying Traffic for OER Using a Prefix List

Before traffic can be measured using OER, it must be identified. Perform this task to use a prefix list to identify the traffic that OER will probe.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network/length* | **permit** *network/length*} [**ge** *ge-value*] [**le** *le-value*]
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network*/*length* | **permit** *network*/*length*} [**ge** *ge-value*] [**le** *le-value*] **Example:** `Router(config)# ip prefix-list TRAFFIC_PFX_LIST seq 10 permit 10.20.21.0/24`	Creates an IP prefix list. • IP prefix lists are used to manually select prefixes for monitoring by the OER master controller. • A master controller can monitor and control an exact prefix (/32), a specific prefix length, or a specific prefix length and any prefix that falls under the prefix length (for example, a /24 under a /16). • A prefix range can also be selected using the **le** keyword with a 32-bit prefix length. • The prefixes specified in the IP prefix list are imported into an OER map using the **match ip address** (OER) command. • The example creates an IP prefix list named TRAFFIC_PFX_LIST that permits prefixes from the 10.20.21.0/24 subnet.
Step 4	**exit** **Example:** `Router(config)# exit`	(Optional) Exits global configuration mode and returns to privileged EXEC mode.

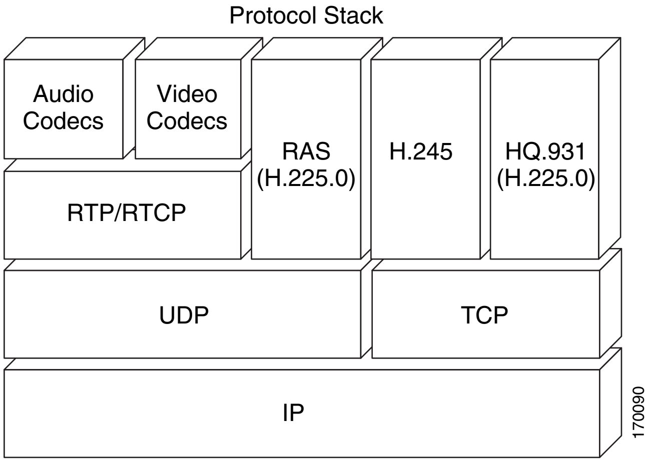
Identifying Voice Traffic to Optimize Using an Access List

Before voice traffic can be measured, it must be identified. Perform this task to use an access list to identify the voice traffic.

IP Protocol Stack for Voice

Voice traffic uses a variety of protocols and streams on the underlying IP network. Figure 2 is a representation of the protocol options available for carrying voice traffic over IP. Most signaling traffic for voice is carried over TCP. Most voice calls are carried over User Datagram Protocol (UDP) and Real-Time Protocol (RTP). You can configure your voice devices to use a specific range of destination port numbers over UDP to carry voice call traffic.

Figure 2 Protocol Stack Options Available for Voice Traffic



SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access list** {**standard** | **extended**} *access-list-name*
4. [*sequence-number*] **permit udp** *source source-wildcard* [*operator* [*port*]] *destination destination-wildcard* [*operator* [*port*]] [**precedence** *precedence*] [**tos** *tos*] [**ttl** *operator value*] [**log**] [**time-range** *time-range-name*] [**fragments**]
5. **exit**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`ip access-list`** `{`**`standard`** `|` **`extended`**`}` *`access-list-name`*
	Example:
	`Router(config)# ip access-list extended VOICE_ACCESS_LIST`
		Defines an IP access list by name.
		• OER supports only named access lists.
		• The example creates an extended IP access list named VOICE_ACCESS_LIST.

Step 4	`[`*`sequence-number`*`]` **`permit udp`** *`source source-wildcard`* `[`*`operator`* `[`*`port`*`]]` *`destination destination-wildcard`* `[`*`operator`* `[`*`port`*`]]` `[`**`precedence`** *`precedence`*`]` `[`**`tos`** *`tos`*`]` `[`**`ttl`** *`operator value`*`]` `[`**`log`**`]` `[`**`time-range`** *`time-range-name`*`]` `[`**`fragments`**`]`
	Example:
	`Router(config-ext-nacl)# permit udp any range 16384 32767 10.20.20.0 0.0.0.15 range 16384 32767`
		Defines the extended access list.
		• Any protocol, port, or other IP packet header value can be specified.
		• The example is configured to identify all UDP traffic ranging from a destination port number of 16384 to 32767 from any source to a destination prefix of 10.20.20.0/24. This specific UDP traffic is to be optimized.

Step 5	**`exit`**
	Example:
	`Router(config)# exit`
		(Optional) Exits global configuration mode and returns to privileged EXEC mode.

Configuring OER Voice Probes with a Target Assignment

After identifying the traffic (in this example, voice traffic identified using an access list) to be optimized, perform this task to configure the OER jitter probes and assign the results of the jitter probes to optimize the identified traffic. In this task, the OER active voice probes are assigned a forced target for OER instead of the usual longest match assigned target. Before configuring the OER jitter probe on the source device, the IP SLAs Responder must be enabled on the target device (the operational target). The IP SLAs Responder is available only on Cisco IOS software-based devices. Start this task at the network device that runs the IP SLAs Responder.

Note The device that runs the IP SLAs Responder does not have to be configured for OER.

Note Policies applied in an OER map do not override global policy configurations.

Prerequisites

Before configuring this task, perform the "Identifying Voice Traffic to Optimize Using an Access List" section on page 7.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla monitor responder**
4. **exit**
5. Move to the network device that is the OER master controller.
6. **enable**
7. **configure terminal**
8. **oer-map** *map-name sequence-number*
9. **match ip address** {**access-list** *access-list-name* | **prefix-list** *prefix-list-name*}
10. **set active probe** *probe-type ip-address* [**target-port** *number*] [**codec** *codec-name*]
11. **set probe frequency** *seconds*
12. **set jitter threshold** *maximum*
13. **set mos threshold** *minimum* **percent** *percent*
14. **set resolve** {**cost priority** *value* | **delay priority** *value* **variance** *percentage* | **jitter priority** *value* **variance** *percentage* | **loss priority** *value* **variance** *percentage* | **mos priority** *value* **variance** *percentage* | **range priority** *value* | **utilization priority** *value* **variance** *percentage*}
15. **set resolve mos priority** *value* **variance** *percentage*
16. **set delay** {**relative** *percentage* | **threshold** *maximum*}
17. **exit**
18. **oer master**
19. **policy-rules** *map-name*
20. **end**
21. **show oer master active-probes forced**
22. **show oer master policy** {*sequence-number* | *policy-name* | **default**}

DETAILED STEPS

	Command or Action	Purpose

Step 1	`enable`
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	`configure terminal`
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

	Command or Action	Purpose

Step 3	**`ip sla monitor responder`**
	Example:
	`Router(config)# ip sla monitor responder`
		Enables the IP SLAs Responder.

Step 4	**`exit`**
	Example:
	`Router(config)# exit`
		Exits global configuration mode and returns to privileged EXEC mode.

Step 5	Move to the network device that is the OER master controller.	—

Step 6	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 7	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 8	**`oer-map`** *`map-name sequence-number`*
	Example:
	`Router(config)# oer-map TARGET_MAP 10`
		Enters OER map configuration mode to configure an OER map to apply policies to selected IP prefixes.
		• *Only one match clause can be configured for each OER map sequence.*
		• Deny sequences are first defined in an IP prefix list and then applied with the **match ip address** (OER) command in Step 9.
		• The example creates an OER map named TARGET_MAP.

Step 9	**`match ip address`** `{`**`access-list`** *`access-list-name`* `|` **`prefix-list`** *`prefix-list-name`*`}`
	Example:
	`Router(config-oer-map)# match ip address access-list VOICE_ACCESS_LIST`
		References an extended IP access list or IP prefix as match criteria in an OER map.
		• Only a single match clause can be configured for each OER map sequence.
		• The example configures the IP access list named VOICE_ACCESS_LIST as match criteria in an OER map. The access list was created in the "Identifying Voice Traffic to Optimize Using an Access List" task.

	Command or Action	Purpose
Step 10	**set active-probe** *probe-type ip-address* [**target-port** *number*] [**codec** *codec-name*] **Example:** `Router(config-oer-map)# set active-probe jitter 10.20.22.1 target-port 2000 codec g729a`	Creates a set clause entry to assign a target prefix for an active probe. • The **echo** keyword is used to specify the target IP address of a prefix to actively monitor using Internet Control Message Protocol (ICMP) echo (ping) messages. • The **jitter** keyword is used to specify the target IP address of a prefix to actively monitor using jitter messages. • The **tcp-conn** keyword is used to specify the target IP address of a prefix to actively monitor using Internet Control Message Protocol (ICMP) echo (ping) messages. • The **udp-echo** keyword is used to specify the target IP address of a prefix to actively monitor using Internet Control Message Protocol (ICMP) echo (ping) messages. • The example creates a set clause entry to specify the target IP address of a prefix and a specific port number to actively monitor using jitter.
Step 11	**set probe frequency** *seconds* **Example:** `Router(config-oer-map)# set probe frequency 10`	Creates a set clause entry to set the frequency of the OER active probe. • The *seconds* argument is used to set the time, in seconds, between the active probe monitoring of the specified IP prefixes. • The example creates a set clause to set the active probe frequency to 10 seconds.
Step 12	**set jitter threshold** *maximum* **Example:** `Router(config-oer-map)# set jitter threshold 20`	Creates a set clause entry to configure the jitter threshold value. • The **threshold** keyword is used to configure the maximum jitter value, in milliseconds. • The example creates a set clause that sets the jitter threshold value to 20 for traffic that is matched in the same OER map sequence.

	Command or Action	Purpose

Step 13	**set mos** {**threshold** *minimum* **percent** *percent*}
	Example:
	`Router(config-oer-map)# set mos threshold 4.0 percent 30`
		Creates a set clause entry to configure the MOS threshold and percentage values used to decide whether an alternate exit is be selected.
		• The **threshold** keyword is used to configure the minimum MOS value.
		• The **percent** keyword is used to configure the percentage of MOS values that are below the MOS threshold.
		• OER calculates the percentage of MOS values below the MOS threshold that are recorded in a five-minute period. If the percentage value exceeds the configured percent value or the default value, the master controller searches for alternate exit links.
		• The example creates a set clause that sets the threshold MOS value to 4.0 and the percent value to 30 percent for traffic that is matched in the same OER map sequence.

Step 14	**set resolve** {**cost priority** *value* | **delay priority** *value* **variance** *percentage* | **jitter priority** *value* **variance** *percentage* | **loss priority** *value* **variance** *percentage* | **mos priority** *value* **variance** *percentage* | **range priority** *value* | **utilization priority** *value* **variance** *percentage*}
	Example:
	`Router(config-oer-map)# set resolve jitter priority 1 variance 10`
		Creates a set clause entry to configure policy priority or resolve policy conflicts.
		• This command is used to set priority for a policy type when multiple policies are configured for the same prefix. When this command is configured, the policy with the highest priority will be selected to determine the policy decision.
		• The **priority** keyword is used to specify the priority value. Configuring the number 1 assigns the highest priority to a policy. Configuring the number 10 assigns the lowest priority.
		• Each policy must be assigned a different priority number.
		• The **variance** keyword is used to set an allowable variance for a user-defined policy. This keyword configures the allowable percentage that an exit link or prefix can vary from the user-defined policy value and still be considered equivalent.
		• Variance cannot be configured for cost or range policies.
		• The example creates set clause that configures the priority for jitter policies to 1 for voice traffic. The variance is configured to allow a 10 percent difference in jitter statistics before a prefix is determined to be out-of-policy.

	Command or Action	Purpose
Step 15	**set resolve mos priority** *value* **variance** *percentage* **Example:** `Router(config-oer-map)# set resolve mos priority 2 variance 15`	Creates a set clause entry to configure policy priority or resolve policy conflicts. • The example creates set clause that configures the priority for MOS policies to 2 for voice traffic. The variance is configured to allow a 15 percent difference in MOS values before a prefix is determined to be out-of-policy. **Note** Only the syntax applicable to this task is used in this example. For more details, see Step 14.

Step 16	**set delay** {**relative** *percentage* | **threshold** *maximum*}
	Example:
	`Router(config-oer-map)# set delay threshold 100`
		Creates a set clause entry to configure the delay threshold.
		• The delay threshold can be configured as a relative percentage or as an absolute value for match criteria.
		• The **relative** keyword is used to configure a relative delay percentage. The relative delay percentage is based on a comparison of short-term and long-term measurements.
		• The **threshold** keyword is used to configure the absolute maximum delay period in milliseconds.
		• The example creates a set clause that sets the absolute maximum delay threshold to 100 milliseconds for traffic that is matched in the same OER map sequence.

Step 17	**exit** **Example:** `Router(config-oer-map)# exit`	Exits OER map configuration mode and returns to global configuration mode.
Step 18	**oer master** **Example:** `Router(config)# oer master`	Enters OER master controller configuration mode to configure a router as a master controller. • A master controller and border router process can be enabled on the same router (for example, in a network that has a single router with two exit links to different service providers). **Note** Only the syntax used in this context is displayed. For more details, see the *Cisco IOS Optimized Edge Routing Command Reference*.
Step 19	**policy-rules** *map-name* **Example:** `Router(config-oer-mc)# policy-rules TARGET_MAP`	Applies a configuration from an OER map to a master controller configuration in OER master controller configuration mode. • Reentering this command with a new OER map name will immediately overwrite the previous configuration. This behavior is designed to allow you to quickly select and switch between predefined OER maps. • The example applies the configuration from the OER map named TARGET_MAP.

	Command or Action	Purpose

Step 20	**end**
	Example:
	`Router(config-oer-mc)# end`
		Exits OER master controller configuration mode and enters privileged EXEC mode.

Step 21	**show oer master active-probes** [**appl** | **forced**] **Example:** `Router# show oer master active-probes forced`	Displays connection and status information about active probes on an OER master controller. • The output from this command displays the active probe type and destination, the border router that is the source of the active probe, the target prefixes that are used for active probing, and whether the probe was learned or configured. • The **appl** keyword is used to filter the output to display information about applications optimized by the master controller. • The **forced** keyword is used to show any forced targets that are assigned. • The example displays connection and status information about the active probes generated for voice traffic configured with a forced target assignment.
Step 22	**show oer master policy** {*sequence-number* | *policy-name* | **default**} **Example:** `Router# show oer master policy TARGET_MAP`	Displays policy settings on an OER master controller. • This command is used to configure an OER map to configure the relative percentage or maximum number of packets that OER will permit to be lost during transmission on an exit link. If packet loss is greater than the user-defined or the default value, the master controller determines that the exit link is out-of-policy. • The *sequence-number* argument is used to display policy settings for the specified OER map sequence. • The *policy-name* argument is used to display policy settings for the specified OER policy map name. • The **default** keyword is used to display only the default policy settings. • The example displays the policy settings configured for the TARGET_MAP policy.

Examples

This example shows output from the **show oer master active-probes forced** command. The output is filtered to display only connection and status information about the active probes generated for voice traffic configured with a forced target assignment.

```
Router# show oer master active-probes forced

OER Master Controller active-probes
Border   = Border Router running this Probe
Policy   = Forced target is configure under this policy
Type     = Probe Type
Target   = Target Address
```

```
TPort     = Target Port
N - Not applicable

The following Forced Probes are running:

Border          State    Policy              Type     Target          TPort
10.20.20.2     ACTIVE    40                  jitter   10.20.22.1      3050
10.20.21.3     ACTIVE    40                  jitter   10.20.22.4      3050
```

What to do Next

For further configuration examples of OER voice traffic optimization, see the "Configuration Examples for OER Voice Traffic Optimization Using Active Probes" section on page 15.

Configuration Examples for OER Voice Traffic Optimization Using Active Probes

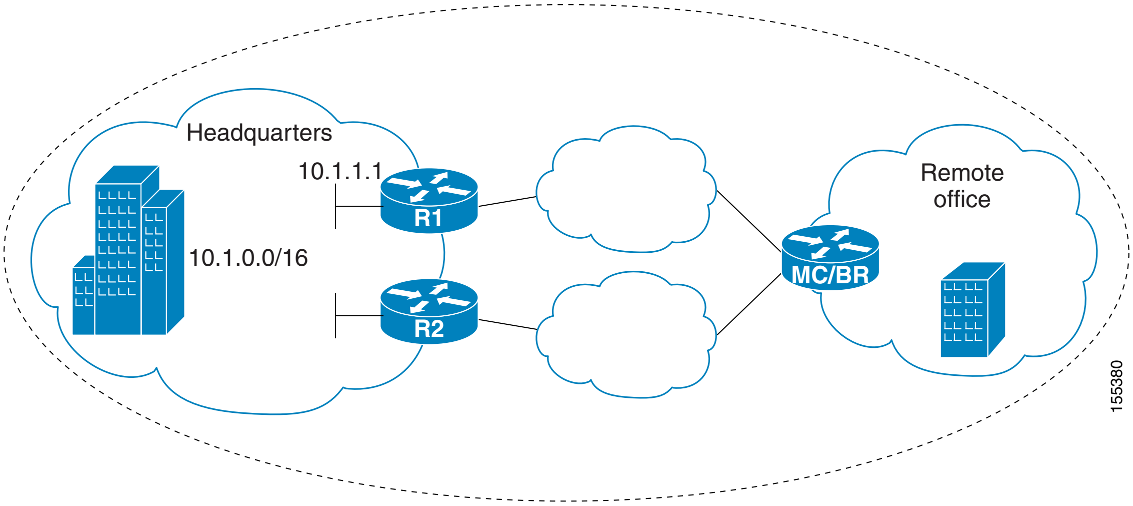
The following examples show both how to use an access list to identify only voice traffic to be optimized by OER and to use a prefix list to identify traffic that includes voice traffic to be optimized by OER.

- Optimizing Only Voice Traffic Using Active Probes: Example, page 15
- Optimizing Traffic (Including Voice Traffic) Using Active Probes: Example, page 16

Optimizing Only Voice Traffic Using Active Probes: Example

Figure 3 shows that voice traffic originating at the remote office and terminating at the headquarters has to be optimized to select the best path out of the remote office network. Degradation in voice (traffic) quality is less likely to be introduced within the network, so probing the edge of the network gives a measurement that is close to probing the final destination.

Figure 3 OER Network Topology Optimizing Voice Traffic Using Active Probes



This configuration optimizes voice traffic to use the best performance path, whereas all other traffic destined to the same network—10.1.0.0/16—will follow the best path as indicated by a traditional routing protocol, for example BGP, that is configured on the device. As part of this optimization, OER will use policy based routing (PBR) to set the best exit link for voice traffic within a device.

The following configuration is performed on the edge router R1 in Figure 3 in the headquarters network to enable the IP SLAs Responder.

```
enable
configure terminal
 ip sla responder
 exit
```

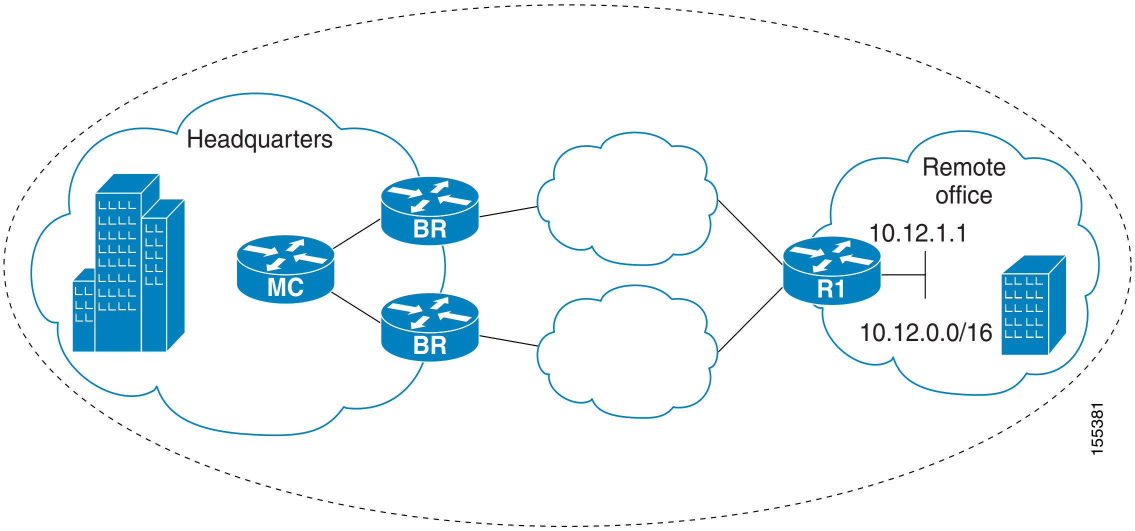
The following configuration is performed on the edge router MC/BR (which is both an OER master controller and border router) in Figure 3 in the remote office network to optimize voice traffic using active probes.

```
enable
configure terminal
ip access-list extended Voice_Traffic
 10 permit udp any 10.1.0.0 0.0.255.255 range  16384 32767
 exit
oer-map Voice_MAP 10
 match ip address access-list Voice_Traffic
 set active-probe jitter 10.1.1.1 target-port 1025 codec g711alaw
 set delay threshold 300
 set mos threshold 3.76 percent 30
 set jitter threshold 15
 set loss relative 5
 resolve mos priority 1
 resolve jitter priority 2
 resolve delay priority 3
 resolve loss priority 4
```

Optimizing Traffic (Including Voice Traffic) Using Active Probes: Example

Figure 4 shows that traffic originating in the headquarters network and destined for the remote office network has to be optimized based on voice traffic metrics. Voice traffic is one of the most important traffic classes that travel from the headquarters to the remote office network, so the voice traffic must be prioritized to be optimized. Degradation in voice packet quality is less likely to be introduced within the network, so probing the edge of the network gives a measurement that is close to probing the final destination.

Figure 4 OER Network Topology for Optimizing All Traffic Using Active Probes



This configuration optimizes all traffic, including voice traffic, destined for the10.12.0.0/16 network. The OER optimization is based on the measurement of voice performance metrics with thresholding values using active probes. As part of the optimization, OER will introduce a BGP or a static route into the headquarters network. For more details about BGP and static route optimization, see the "Using OER to Control Traffic Classes and Verify the Route Control Changes" module.

The following configuration is performed on router R1 in Figure 4 in the remote office network to enable the IP SLAs Responder.

```
enable
configure terminal
 ip sla responder
 exit
```

The following configuration is performed on one of the BR routers in Figure 4 in the headquarters network to optimize all traffic (including voice traffic) using active probes.

```
enable
configure terminal
 ip prefix-list All_Traffic_Prefix permit 10.12.0.0/16
 oer-map Traffic_MAP 10
 match ip address prefix-list All_Traffic_Prefix
 set active-probe jitter 10.12.1.1 target-port 1025 codec g711alaw
! port 1025 for the target probe is an example.
 set delay threshold 300
 set mos threshold 3.76 percent 30
 set jitter threshold 15
 set loss relative 5
 resolve mos priority 1
 resolve jitter priority 2
 resolve delay priority 3
 resolve loss priority 4
```

Where to Go Next

This document describes a specific implementation of OER and presumes that you are familiar with the OER technology. If you want to review more information about OER, proceed to the Cisco IOS Optimized Edge Routing Overview module, followed by the Setting Up OER Network Components module. If you have set up your OER components, you should read through the other modules in the following list:

- Using OER to Profile the Traffic Classes
- Measuring the Traffic Class Performance and Link Utilization Using OER
- Configuring and Applying OER Policies
- Using OER to Control Traffic Classes and Verify the Route Control Changes

After you understand the various OER phases, review the OER solutions modules that are listed under "Related Documents" section on page 18.

Additional References

The following sections provide references related to optimizing voice traffic using OER active probes.

Related Documents

Related Topic	Document Title
Cisco OER technology overview	"Cisco IOS Optimized Edge Routing Overview" module
Concepts and configuration tasks required to set up OER network components.	"Setting Up OER Network Components" module
OER solution module: configuring VPN IPsec/GRE tunnel interfaces as OER-managed exit links.	"Configuring VPN IPsec/GRE Tunnel Interfaces As OER-Managed Exit Links" module
Cisco OER commands: complete command syntax, command mode, command history, defaults, usage guidelines and examples	*Cisco IOS Optimized Edge Routing Command Reference*
IP Routing Protocol commands	*Cisco IOS IP Routing Protocols Command Reference*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for OER Voice Traffic Optimization Using Active Probes

Table 1 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.4(6)T, 12.2(33)SRB, or a later release appear in the table.

For information on a feature in this technology that is not documented here, see the "Cisco IOS Optimized Edge Routing Feature Roadmap."

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 ***Feature Information for OER Voice Traffic Optimization Using Active Probes***

Feature Name	Releases	Feature Information
OER Voice Traffic Optimization	12.4(6)T 12.2(33)SRB	The OER Voice Traffic Optimization feature provides support for outbound optimization of voice traffic based on the voice metrics, jitter and Mean Opinion Score (MOS). Jitter and MOS are important quantitative quality metrics for voice traffic and these voice metrics are measured using OER active probes. The following commands were introduced or modified by this feature: **active-probe**, **jitter**, **mos**, **resolve**, **set active-probe**, **set jitter**, **set mos**, **set probe**, **set resolve**, **show oer master active-probes**, **show oer master policy**, and **show oer master prefix**.

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

Configuring VPN IPsec/GRE Tunnel Interfaces As OER-Managed Exit Links

First Published: August 14, 2006
Last Updated: July 31, 2006

This module documents an Optimized Edge Routing (OER) solution that describes how to configure IP security (IPsec)/Generic Routing Encapsulation (GRE) tunnel interfaces as OER-managed exit links. The VPN IPsec/GRE Tunnel Optimization solution was introduced in Cisco IOS Release 12.3(11)T, and only network-based IPsec Virtual Private Networks (VPNs) are supported.

OER provides automatic route optimization and load distribution for multiple connections between networks. OER is an integrated Cisco IOS solution that allows you to monitor IP traffic flows and then define policies and rules based on prefix performance, link load distribution, link bandwidth monetary cost, and traffic type. OER provides active and passive monitoring systems, dynamic failure detection, and automatic path correction. Deploying OER enables intelligent load distribution and optimal route selection in an enterprise network.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Configuring VPN IPsec/GRE Tunnel Interfaces As OER-Managed Exit Links" section on page 19.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Contents

Prerequisites for Configuring VPN IPsec/GRE Tunnel Interfaces As OER-Managed Exit Links

- Before implementing VPN IPsec/GRE tunnel interfaces as OER-managed exit links you need to understand how to configure a basic OER-managed network. See the "Cisco IOS Optimized Edge Routing Overview" and "Setting Up OER Network components" modules for more details. For a list of other OER configuration modules, see the "Where to Go Next" section on page 18 and the "Related Documents" section on page 18.
- Cisco Express Forwarding (CEF) must be enabled on all participating routers.
- Routing protocol peering or static routing is configured in the OER-managed network.
- Standard Cisco OER border router and master controller configurations are completed.

Restrictions for Configuring VPN IPsec/GRE Tunnel Interfaces As OER-Managed Exit Links

Cisco IOS OER supports the optimization of prefixes that are routed over IPsec/GRE tunnel interfaces. Only GRE and multipoint GRE VPN tunnels are supported.

Information About Configuring VPN IPsec/GRE Tunnel Interfaces As OER-Managed Exit Links

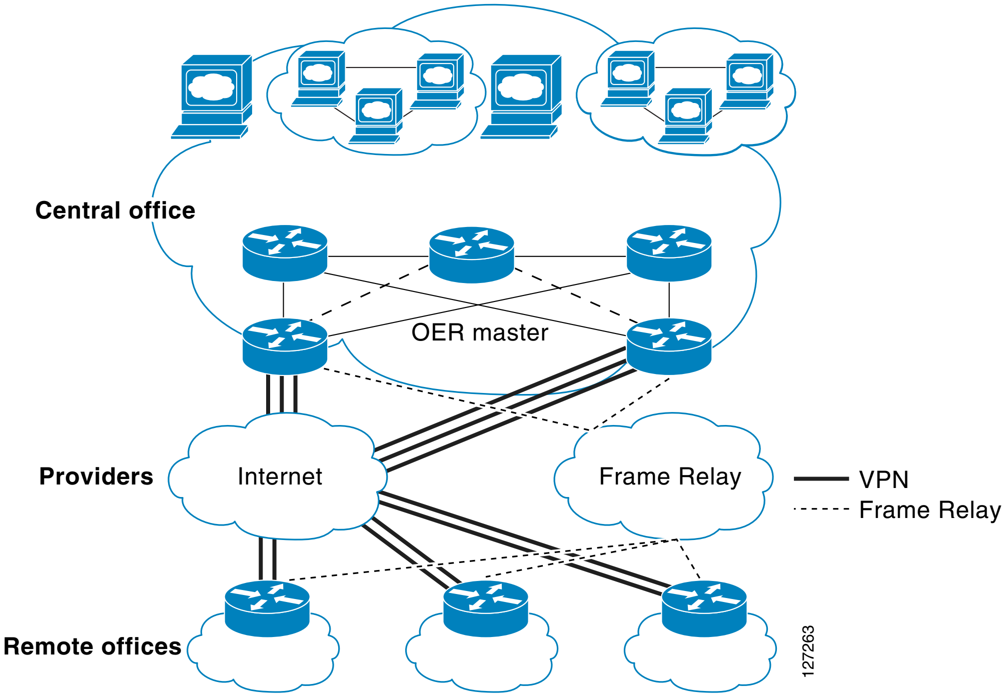
To configure VPN IPsec/GRE tunnel interfaces as OER-managed exit links you should understand the following concepts:

- VPN IPsec/GRE Tunnel Interface Optimization, page 3
- Protection of Route Prefixes with IPsec over GRE Tunnels, page 4

VPN IPsec/GRE Tunnel Interface Optimization

Cisco IOS OER supports the optimization of prefixes that are routed over IPsec/GRE tunnel interfaces. The VPN tunnel interface is configured as OER external interfaces on the master controller. Figure 1 shows an OER-managed network that is configured to optimize VPN traffic. Cisco IOS OER is deployed at the central office and remote offices.

Figure 1 Cisco IOS OER Network Optimized for VPN Routing



This enhancement allows you to configure two-way VPN optimization. A master controller and border router process are enabled on each side of the VPN. Each site maintains a separate master controller database. VPN routes can be dynamically learned through the tunnel interfaces or can be configured. Prefix and exit link policies are configured for VPN prefixes through a standard Cisco IOS OER configuration.

Protection of Route Prefixes with IPsec over GRE Tunnels

The IPsec-to-GRE model allows a service provider to provide VPN services over the IP backbone. Both the central and remote VPN clients terminate according to the IPsec-to-IPsec model. Prefixes are encapsulated using GRE tunnels. The GRE packet is protected by IPsec. The encapsulated prefixes are forwarded from the central VPN site to a customer headend router that is the other endpoint for GRE. The IPsec-protected GRE packets provide secure connectivity across the IP backbone of the service provider network.

For more information about configuring IPsec over GRE tunnels, see the *Dynamic Multipoint IPsec VPNs (Using Multipoint GRE/NHRP to Scale IPsec VPNs)* document published at the following URL:

http://www.cisco.com/en/US/tech/tk583/tk372/technologies_white_paper09186a008018983e.shtml

How to Configure VPN IPsec/GRE Tunnel Interfaces As OER-Managed Exit Links

This section contains the following task:

- Configuring OER to Monitor and Control IPsec VPN Prefixes over GRE Tunnels, page 4

Configuring OER to Monitor and Control IPsec VPN Prefixes over GRE Tunnels

Perform this task to configure the IPsec VPN configuration over GRE tunnels. Initially the IPsec VPN is configured on a border router, and the tunnel interface is configured as an OER-managed external interface on the master controller. In this task an IKE policy is defined, a transform set is configured, a crypto profile and a crypto map are defined, and a GRE tunnel is configured.

The GRE tunnel and IPsec protection in this task are configured on the border router. The configuration steps in this task show how to configure a single tunnel. At least two tunnels must be configured on border routers in an OER-managed network. The IPsec configuration must be applied at each tunnel endpoint (the central and remote site).

Configuration of GRE Tunnel Interfaces As OER-Managed Exit Links

GRE tunnel interfaces on the border routers are configured as OER external interfaces on the master controller. At least two external tunnel interfaces must be configured on separate physical interfaces in an OER-managed network. These interfaces can be configured on a single border router or multiple border routers. Internal interfaces are configured normally using a physical interface that is on the border router and is reachable by the master controller.

Restrictions

Cisco IOS OER supports only IPsec/GRE VPNs. No other VPN types are supported.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **crypto ipsec security-association lifetime** {**seconds** *seconds* | **kilobytes** *kilobytes*}
4. **crypto ipsec transform-set** *transform-set-name transform1* [*transform2*] [*transform3*] [*transform4*]
5. **mode** [**tunnel** | **transport**]
6. **exit**
7. **crypto map** *map-name seq-num* [**ipsec-isakmp**]
8. **set peer** {*host-name* [**dynamic**] [**default**] | *ip-address* [**default**]}
9. **set transform-set** *transform-set-name* [*transform-set-name2...transform-set-name6*]
10. **match address** [*access-list-id* | *name*]
11. **exit**
12. **crypto ipsec profile** *name*
13. **set transform-set** *transform-set-name* [*transform-set-name2...transform-set-name6*]
14. **exit**
15. **crypto map** *map-name* **local-address** *interface-id*
16. **crypto isakmp key** *encryption-level key-string* {**address** *peer-address* [*mask*] | **hostname** *name*} [**no-xauth**]
17. **crypto isakmp keepalive** *seconds* [*retries*] [**periodic** | **on-demand**]
18. **crypto isakmp policy** *priority*
19. **encryption** {**des** | **3des** | **aes** | **aes 192** | **aes 256**}
20. **authentication** {**rsa-sig** | **rsa-encr** | **pre-share**}
21. **exit**
22. **interface** *type number* [*name-tag*]
23. **ip address** *ip-address mask* [**secondary**]
24. **crypto map** *map-name* [**redundancy** *standby-name*]
25. exit
26. **interface** *type number* [*name-tag*]
27. **ip address** *ip-address mask* [**secondary**]
28. **keepalive** [*period* [*retries*]]
29. **bandwidth** {*kbps* | **inherit** [*kbps*]}
30. **tunnel mode gre ip**
31. **tunnel source** {*ip-address* | *interface-type interface-number*}
32. **tunnel destination** {*host-name* | *ip-address*}
33. **tunnel protection ipsec profile** *name* [**shared**]
34. **exit**
35. **ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [**dhcp**] [*distance*] [*name*] [**permanent**] [**tag** *tag*]

36. **access-list** *access-list-number* [**dynamic** *dynamic-name* [**timeout** *minutes*]] {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**tos** *tos*] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**]

37. end

DETAILED STEPS

	Command or Action	Purpose

Step 1	`enable`
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	`configure terminal`
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**crypto ipsec security-association lifetime** {**seconds** *seconds* | **kilobytes** *kilobytes*}
	Example:
	`Router(config)# crypto ipsec security-association lifetime kilobytes 530000000`
		Sets global lifetime values used when negotiating IPsec security associations.
		• The example sets volume of traffic, in kilobytes, that can pass between IPsec peers for this security association.

Step 4	**crypto ipsec transform-set** *transform-set-name transform1* [*transform2*] [*transform3*] [*transform4*] **Example:** `Router(config)# crypto ipsec transform-set VPN_1 esp-des esp-3des esp-sha-hmac`	Enters crypto transform configuration mode to create or modify a transform set—an acceptable combination of security protocols and algorithms. • The example specifies 56-bit Data Encryption Standard (DES), 168-bit DES, or Secure Hash Algorithm (SHA) for authentication.
Step 5	**mode** [**tunnel** | **transport**] **Example:** `Router(cfg-crypto-trans)# mode transport`	Sets the mode for the transform set. • The example sets the mode to transport. The default mode is tunnel. Under tunnel mode, the entire packet is protected. Under transport mode, only the payload is protected. Encapsulation is performed by GRE.

Step 6	`exit`
	Example:
	`Router(cfg-crypto-trans)# exit`
		Exits crypto transform configuration mode and enters global configuration mode.

Step 7	**crypto map** *map-name seq-num* [**ipsec-isakmp**] **Example:** `Router(config)# crypto map TUNNEL 10 ipsec-isakmp`	Enters crypto map configuration mode to create or modify a crypto map. • The example creates a crypto map named TUNNEL and configures IKE to establish the security association.

	Command or Action	Purpose

Step 8	**set peer** {*host-name* [**dynamic**] [**default**] | *ip-address* [**default**]}
	Example:
	`Router(config-crypto-map)# set peer 10.4.9.81`
		Specifies the IPsec peer in the crypto map entry.

Step 9	**set transform-set** *transform-set-name* [*transform-set-name2...transform-set-name6*] **Example:** `Router(config-crypto-map)# set transform-set VPN_1`	Specifies which transform sets can be used with the crypto map entry. • Specifies the transform set VPN_1, which was configured in Step 4.
Step 10	**match address** [*access-list-id* | *name*] **Example:** `Router(config-crypto-map)# match address 100`	Specifies an extended access list to define IPsec peers for the crypto map entry. • The access list is defined in Step 36.
Step 11	**exit** **Example:** `Router(config-crypto-map)# exit`	Exits crypto map configuration mode and enters global configuration mode.
Step 12	**crypto ipsec profile name** **Example:** `Router(config)# crypto ipsec profile OER`	Defines the IPsec parameters that are to be used for IPsec encryption between two IPsec routers and enters IPsec profile configuration mode. • The example creates a profile named OER.
Step 13	**set transform-set** *transform-set-name* [*transform-set-name2...transform-set-name6*] **Example:** `Router(ipsec-profile)# set transform-set VPN_1`	Specifies which transform sets can be used with the crypto map entry. • Specifies the transform set VPN_1, which was configured in Step 4.
Step 14	**exit** **Example:** `Router(ipsec-profile)# exit`	Exits IPsec profile configuration mode and enters global configuration mode.
Step 15	**crypto map** *map-name* **local-address** *interface-id* **Example:** `Router(config)# crypto map TUNNEL local-address FastEthernet 0/0`	Attaches a defined crypto map to the specified interface. • The example attaches the crypto map named TUNNEL to interface FastEthernet 0/0.
Step 16	**crypto isakmp key** *encryption-level key-string* {**address** *peer-address* [*mask*] | **hostname** *name*} [**no-xauth**] **Example:** `Router(config)# crypto isakmp key 0 CISCO address 10.4.9.81 no-xauth`	Creates the preshared authentication key. • The example configures encryption level 0 and configures the router to not prompt the IPsec peer for extended authentication. However, any encryption level or authentication level can be specified.

	Command or Action	Purpose
Step 17	**crypto isakmp keepalive** *seconds* [*retries*] [**periodic** | **on-demand**] **Example:** `Router(config)# crypto isakmp keepalive 10`	Allows the gateway to send dead peer detection (DPD) messages to the peer.
Step 18	**crypto isakmp policy** *priority* **Example:** `Router(config)# crypto isakmp policy 1`	Defines an Internet Key Exchange (IKE) policy and enters ISAKMP policy configuration mode.

Step 19	**encryption** {**des** | **3des** | **aes** | **aes 192** | **aes 256**}
	Example:
	`Router(config-isakmp)# encryption 3des`
		Specifies the encryption algorithm within the IKE policy.
		• The example specifies 168-bit DES encryption.

Step 20	**authentication** {**rsa-sig** | **rsa-encr** | **pre-share**}
	Example:
	`Router(config-isakmp)# authentication pre-share`
		Specifies the authentication method within the IKE policy.
		• The example specifies that a preshared key will be used.

Step 21	**exit** **Example:** `Router(config-isakmp)# exit`	Exits ISAKMP policy configuration mode and enters global configuration mode.
Step 22	**interface** *type number* [*name-tag*] **Example:** `Router(config)# interface FastEthernet0/0`	Configures an interface type and enters interface configuration mode. • The physical interface is defined in this step.
Step 23	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 10.4.9.14 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 24	**crypto map** *map-name* [**redundancy** *standby-name*] **Example:** `Router(config-if)# crypto map TUNNEL`	Applies the crypto map set to the interface. • The example specifies the crypto map named TUNNEL, which was defined in Step 7.
Step 25	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 26	**interface** *type number* [*name-tag*] **Example:** `Router(config)# interface Tunnel0`	Configures an interface type and enters interface configuration mode. • The tunnel interface is defined in this step.

	Command or Action	Purpose
Step 27	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(config-if) ip address 10.100.2.1 255.255.0.0`	Sets a primary or secondary IP address for an interface.
Step 28	**keepalive** [*period* [*retries*]] **Example:** `Router(config-if) keepalive 30 5`	Enables keepalive packets and specifies the number of times that the Cisco IOS software tries to send keepalive packets without a response before bringing down the interface or before bringing the tunnel protocol down for a specific interface.

Step 29	**bandwidth** {*kbps* | **inherit** [*kbps*]}
	Example:
	`Router(config-if)# bandwidth 500`
	`Router(config-if)# bandwidth inherit`
		Sets and communicates the current bandwidth value for an interface to higher-level protocols.

Step 30	**tunnel mode gre ip** **Example:** `Router(config-if)# tunnel mode gre ip`	Sets the encapsulation mode for the tunnel interface. **Note** Only partial syntax is shown here. For more details, see the *Cisco IOS Interface and Hardware Component Command Reference*, 12.4T.

Step 31	**tunnel source** {*ip-address* | *interface-type interface-number*}
	Example:
	`Router(config-if)# tunnel source 10.4.9.14`
		Sets the source address for a tunnel interface.
		• The source interface in the example was defined in Step 22. The interface name or IP address can be specified.

Step 32	**tunnel destination** {*host-name* | *ip-address*}
	Example:
	`Router(config-if)# tunnel destination 10.4.9.81`
		Specifies the destination for a tunnel interface.
		• The IP address of the physical interface where the remote tunnel end point is attached is configured in this step.

Step 33	**tunnel protection ipsec profile** *name* [**shared**] **Example:** `Router(config-if)# tunnel protection ipsec profile OER`	Associates the tunnel interface with the IPsec profile. • The IPsec profile named OER that is configured in the example was defined in Step 19.
Step 34	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 35	**ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [**dhcp**] [*distance*] [*name*] [**permanent**] [**tag** *tag*] **Example:** `Router(config)# ip route 10.2.2.2 255.255.255.255 Tunnel0`	Establishes a static route. • A default route is configured for the tunnel destination host or network.

	Command or Action	Purpose
Step 36	**access-list** *access-list-number* [**dynamic** *dynamic-name* [**timeout** *minutes*]] {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**tos** *tos*] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Router(config)# access-list 100 permit gre host 10.4.9.14 host 10.4.9.81`	Creates or configures an extended IP access list. • An extended access list is defined to permit only the GRE hosts. • The access list in this example is referenced in the **match address** statement in Step 10.

Step 37	**end**
	Example:
	`Router(config)# end`
		Exits global configuration mode and enters privileged EXEC mode.

Configuration Examples for Configuring VPN IPsec/GRE Tunnel Interfaces As OER-Managed Exit Links

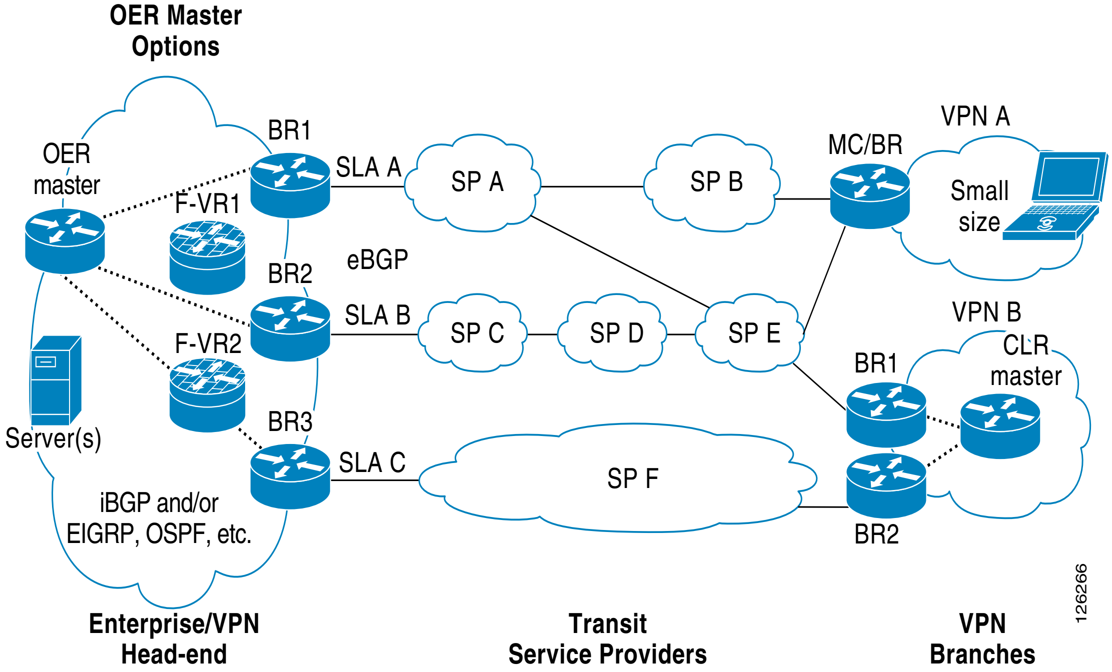
This section contains the following example:

- Configuring OER to Monitor and Control GRE/IPsec VPN Prefixes: Example, page 10

Configuring OER to Monitor and Control GRE/IPsec VPN Prefixes: Example

Figure 2 shows a central VPN site and two remote VPN sites. VPN peering is established through the service provider clouds. An OER-managed network is configured at each site where Cisco IOS OER configuration is applied independently. Each site has a separate master controller and border router process, and each site maintains a separate master controller database.

Figure 2 VPN Sites Controlled by OER-Managed Networks



Two GRE tunnels are configured between each remote site and the central site. VPN prefixes are encapsulated in GRE tunnels, which in turn are protected by IPsec encryption. The examples in this section show the configuration for the central VPN site, VPN A, and VPN B.

Central VPN Configuration: OER Master Controller

The central VPN site peers with VPN A and VPN B. A separate policy is defined for each site using an OER map. For VPN A prefixes, a delay policy of 80 ms is configured and out-of-policy prefixes are moved to the first in-policy exit. For VPN B prefixes, a delay policy of 40 ms and a relative loss policy are configured, and out-of-policy prefixes are moved to the best available exit.

```
key chain OER
 key 1
  key-string CISCO
!
oer master
 logging
 border 10.4.9.6 key-chain OER
  interface Ethernet 0/0 external
  interface Ethernet 0/1 internal
!
 border 10.4.9.7 key-chain OER
  interface Ethernet 0/0 external
  interface Ethernet 0/1 internal
!
 mode route control
 mode monitor both
 exit
!
ip prefix VPN A permit 10.4.9.25
oer-map VPNA
 match ip address prefix-list VPNB
 set delay 800
 set mode select-exit good
 exit
```

```
!
ip prefix VPNB permit 10.4.9.254
oer-map VPNB
 match ip address prefix-list VPNC
 set delay 400
 set loss relative 100
 set resolve loss priority 1 variance 10
 set mode select-exit best
 end
```

Central VPN Configuration: BR1

The following example, starting in global configuration mode, shows the central VPN configuration for BR1:

```
key chain OER
 key 1
  key-string CISCO
!
oer border
 local serial 0/1
 master 10.4.9.4 key-chain OER
!
ip route 10.70.1.0 255.255.255.0
!
route-map REDISTRIBUTE_STATIC
 match tag 5000
 set metric -10
 exit
!
router eigrp 1
 network 10.70.0.0 0.0.0.255
 redistribute static route-map REDISTRIBUTE_STATIC
 exit
!
crypto ipsec security-association lifetime kilobytes 530000000
crypto ipsec security-association lifetime second 14400
crypto ipsec transform-set VPN_1 esp-3des esp-sha-hmac
 mode transport
 exit
!
crypto map TUNNEL 10 ipsec-isakmp
 set peer 10.4.9.81
 set transform-set VPN_1
 match address 100
!
crypto ipsec profile OER
 set transform-set VPN_1
 exit
crypto map TUNNEL local-address Ethernet 0/0
!
crypto isakmp key 0 CISCO address 10.4.9.81 no-xauth
crypto isakmp keepalive 10
crypto isakmp policy 1
 encryption 3des
 authentication pre-share
 exit
!
interface Ethernet0/0
 ip address 10.4.9.14 255.255.255.0
 crypto map TUNNEL
 exit
!
interface Tunnel0
```

```
 ip address 10.100.2.1 255.255.0.0
 keepalive 30 5
 bandwidth 500
 bandwidth inherit
 tunnel mode gre ip
 tunnel source 10.4.9.14
 tunnel destination 10.4.9.81
 tunnel protection ipsec profile OER
 exit
```

Central VPN Configuration: BR2

The following example, starting in global configuration mode, shows the central VPN configuration of BR2:

```
key chain OER
 key 1
  key-string CISCO
!
oer border
 local Ethernet 0/1
 master 10.4.9.4 key-chain OER
!
ip route 10.70.1.0 255.255.255.0
!
route-map REDISTRIBUTE_STATIC
 match tag 5000
 set metric -10
 exit
!
router eigrp 1
 network 10.70.0.0 0.0.0.255
 redistribute static route-map REDISTRIBUTE_STATIC
!
crypto ipsec security-association lifetime kilobytes 530000000
crypto ipsec security-association lifetime second 14400
crypto ipsec transform-set VPN_1 esp-3des esp-sha-hmac
 mode transport
 exit
!
crypto map TUNNEL 10 ipsec-isakmp
 set peer 10.4.9.82
 set transform-set VPN_1
 match address 100
!
crypto ipsec profile OER
 set transform-set VPN_1
 exit
crypto map TUNNEL local-address Ethernet 0/0
!
crypto isakmp key 0 CISCO address 10.4.9.82 no-xauth
crypto isakmp keepalive 10
crypto isakmp policy 1
 encryption 3des
 authentication pre-share
 exit
!
interface Ethernet0/0
 ip address 10.4.9.15 255.255.255.0
 crypto map TUNNEL
 exit
!
interface Tunnel0
 ip address 10.100.2.2 255.255.0.0
 keepalive 30 5
 bandwidth 500
 bandwidth inherit
 tunnel mode gre ip
 tunnel source 10.4.9.15
 tunnel destination 10.4.9.82
 tunnel protection ipsec profile OER
 end
```

Central VPN Configuration: Internal Peers

The following example shows an EIGRP routing process created to establish peering with the border routers and internal peers:

```
router eigrp 1
 network 10.50.1.0 0.0.0.255
 redistribute static route-map REDISTRIBUTE_STATIC
 end
```

VPN A Configuration: MC/BR

The following configuration example, starting in global configuration mode, shows the configuration of VPN A. VPN A is a remote site that is configured for a small office home office (SOHO) client. A single router is deployed. This router peers with service provider B and service provider E. No Interior Gateway Protocol (IGP) is deployed at this network; only a static route is configured to the remote tunnel endpoint at the central site. A delay policy, a loss policy, and optimal exit link selection are configured so that traffic is always routed through the ISP with the lowest delay time and lowest packet loss. A resolve policy is configured to configure loss to have the highest priority. Neither the physical interface configuration nor the router IGP peering configurations are shown in this example.

```
key chain BR1
 key 1
  key-string CISCO
!
```

Note The local border router process is enabled. Because the border router and master controller process is enabled on the same router, a loopback interface (192.168.0.1) is configured as the local interface.

```
oer border
 local Loopback0
 master 192.168.0.1 key-chain BR1
!
oer master
 learn
 delay
 mode route control
 delay threshold 100
 loss relative 200
 periodic 300
 mode select-exit good
resolve loss priority 1 variance 20
resolve delay priority 2 variance 10
!
 border 192.168.0.1 key-chain BR1
  interface Serial0/0 internal
  interface Tunnel0 external
  interface Tunnel0 external
  exit
!
crypto ipsec security-association lifetime kilobytes 530000000
crypto ipsec security-association lifetime second 14400
crypto ipsec transform-set VPN_1 esp-3des esp-sha-hmac
```

```
 mode transport
 exit
!
crypto map TUNNEL 10 ipsec-isakmp
 set peer 10.4.9.81
 set transform-set VPN_1
 match address 100
!
crypto ipsec profile OER
 set transform-set VPN_1
 exit
crypto map TUNNEL local-address Ethernet 0/0
!
crypto isakmp key 0 CISCO address 10.4.9.81 no-xauth
crypto isakmp keepalive 10
crypto isakmp policy 1
 encryption 3des
 authentication pre-share
 exit
!

interface Ethernet0/0
 ip address 10.4.9.14 255.255.255.0
 crypto map TUNNEL
 exit
!
interface Tunnel0
 ip address 10.100.2.1 255.255.0.0
 keepalive 30 5
 bandwidth 500
 bandwidth inherit
 tunnel mode gre ip
 tunnel source 10.4.9.14
 tunnel destination 10.4.9.81
 tunnel protection ipsec profile OER
 exit
!
```

Note A single tunnel configuration is show in this example. Two tunnels are required to configure VPN optimization.

VPN B Configuration: OER Master Controller

The following example, starting in global configuration mode, shows the master controller configuration in VPN B. Load distribution and route control mode are enabled. Out-of-policy prefixes are configured to be moved to the first in-policy exit.

```
key chain OER
 key 1
  key-string CISCO
!
oer master
 logging
 border 10.4.9.6 key-chain OER
  interface Ethernet 0/0 external
  interface Ethernet 0/1 internal
!
 border 10.4.9.7 key-chain OER
  interface Ethernet 0/0 external
  interface Ethernet 0/1 internal
!
mode route control
mode select-exit good
max-range utilization
!
 learn
  delay
  end
```

VPN B Configuration: BR1

The following example, starting in global configuration mode, shows the VPN B configuration for BR1:

```
key chain OER
 key 1
  key-string CISCO
!
oer border
 local Ethernet 0/1
 master 10.4.9.4 key-chain OER
!
route-map REDISTRIBUTE_STATIC
 match tag 5000
 set metric -10
 exit
!
router rip
 network 10.60.1.0
 redistribute static route-map REDISTRIBUTE_STATIC
 end
!
crypto ipsec security-association lifetime kilobytes 530000000
crypto ipsec security-association lifetime second 14400
crypto ipsec transform-set VPN_1 esp-3des esp-sha-hmac
 mode transport
 exit
!
crypto map TUNNEL 10 ipsec-isakmp
 set peer 10.4.9.82
 set transform-set VPN_1
 match address 100
!
crypto ipsec profile OER
 set transform-set VPN_1
 exit
crypto map TUNNEL local-address Ethernet 0/0
!
crypto isakmp key 0 CISCO address 10.4.9.82 no-xauth
crypto isakmp keepalive 10
crypto isakmp policy 1
 encryption 3des
 authentication pre-share
 exit
!
interface Ethernet0/0
 ip address 10.4.9.15 255.255.255.0
 crypto map TUNNEL
 exit
!
interface Tunnel0
 ip address 10.100.2.2 255.255.0.0
 keepalive 30 5
 bandwidth 500
 bandwidth inherit
 tunnel mode gre ip
 tunnel source 10.4.9.15
 tunnel destination 10.4.9.82
 tunnel protection ipsec profile OER
 end
```

VPN B Configuration: BR2

The following example, starting in global configuration mode, shows the VPN B configuration for BR2:

```
key chain OER
 key 1
  key-string CISCO
!
oer border
 local Ethernet 0/1
 master 10.4.9.4 key-chain OER
 exit
!
route-map REDISTRIBUTE_STATIC
 match tag 5000
 set metric -10
 exit
!
router rip
 network 10.60.1.0
 redistribute static route-map REDISTRIBUTE_STATIC
 exit
!
crypto ipsec security-association lifetime kilobytes 530000000
crypto ipsec security-association lifetime second 14400
crypto ipsec transform-set VPN_1 esp-3des esp-sha-hmac
 mode transport
 exit
!
crypto map TUNNEL 10 ipsec-isakmp
 set peer 10.4.9.82
 set transform-set VPN_1
 match address 100
!
crypto ipsec profile OER
 set transform-set VPN_1
 exit
crypto map TUNNEL local-address Ethernet 0/0
!
crypto isakmp key 0 CISCO address 10.4.9.82 no-xauth
crypto isakmp keepalive 10
crypto isakmp policy 1
 encryption 3des
 authentication pre-share
 exit
!
interface Ethernet0/0
 ip address 10.4.9.15 255.255.255.0
 crypto map TUNNEL
 exit
!
interface Tunnel0
 ip address 10.100.2.2 255.255.0.0
 keepalive 30 5
 bandwidth 500
 bandwidth inherit
 tunnel mode gre ip
 tunnel source 10.4.9.15
 tunnel destination 10.4.9.82
 tunnel protection ipsec profile OER
 end
```

VPN B Configuration: Internal Peers

The following example shows a Routing Information Protocol (RIP) routing process created to establish peering with the border routers and internal peers:

```
router rip
 network 10.60.1.0
 end
```

Where to Go Next

This document describes a specific implementation of OER and presumes that you are familiar with the OER technology. If you want to review more information about OER, proceed to the Cisco IOS Optimized Edge Routing Overview module, followed by the Setting Up OER Network Components module. To learn more about the other OER phases, read through the other modules in the following list:

- Using OER to Profile the Traffic Classes
- Measuring the Traffic Class Performance and Link Utilization Using OER
- Configuring and Applying OER Policies
- Using OER to Control Traffic Classes and Verify the Route Control Changes

After you understand the various OER phases you may want to review other OER Solutions modules that are listed under "Related Documents" section on page 18.

Additional References

The following sections provide references related to configuring VPN IPsec/GRE tunnel interfaces as OER-managed exit links.

Related Documents

Related Topic	Document Title
Cisco OER technology overview	"Cisco IOS Optimized Edge Routing Overview" module
Concepts and configuration tasks required to set up OER network components.	"Setting Up OER Network Components" module
OER solution module: voice traffic optimization using OER active probes.	"OER Voice Traffic Optimization Using Active Probes" module
Cisco OER commands: complete command syntax, command mode, command history, defaults, usage guidelines and examples	*Cisco IOS Optimized Edge Routing Command Reference*
IP Routing Protocol commands	*Cisco IOS IP Routing Protocols Command Reference*
Key Chain Authentication: information about authentication key configuration and management in Cisco IOS software	"Managing Authentication Keys" section of the "Configuring IP Routing Protocol-Independent Features" chapter in the *Cisco IOS IP Routing Protocols Configuration Guide*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Configuring VPN IPsec/GRE Tunnel Interfaces As OER-Managed Exit Links

Table 1 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.3(11)T or a later release appear in the table.

For information on a feature in this technology that is not documented here, see the "Cisco IOS Optimized Edge Routing Feature Roadmap."

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for VPN IPsec/GRE Tunnel Interface Optimization

Feature Name	Releases	Feature Information
VPN IPsec/GRE Tunnel Optimization	12.3(11)T	Introduces the ability to configure IPsec/GRE tunnel interfaces as OER-managed exit links.

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.