Using OER to Control Traffic Classes and Verify the Route Control Changes

Last Updated: October 10, 2011

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

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

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

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 Where to Go Next 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

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.

OERTrafficClassControlTechniques

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, alter BGP attributes, or introduce policy-based routing using a route map 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.

The figure below 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***
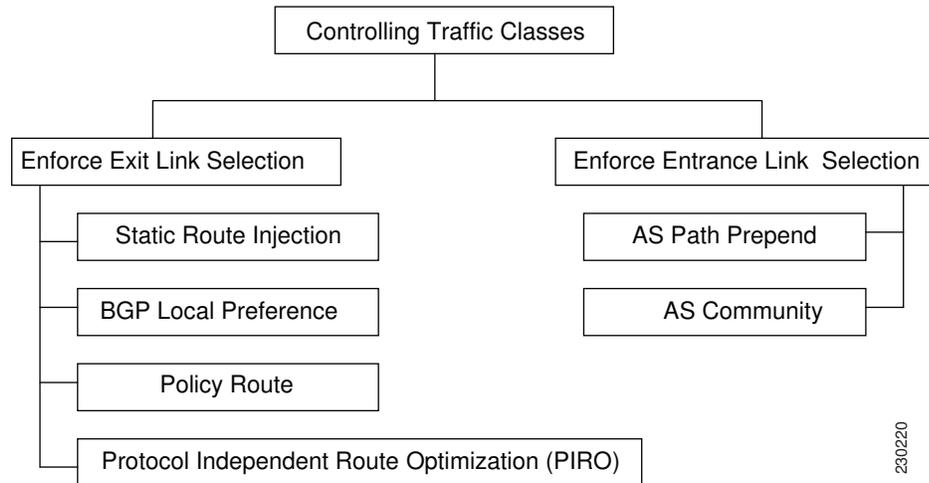
For more details about the different traffic class control techniques that can be implemented, see the following sections:

- 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

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.

EIGRP Route Control

In Cisco IOS Release 15.0(1)M, 12.2(33)SRE, and later releases, the PfR EIGRP mGRE DMVPN Hub-and-Spoke Support feature introduced PfR route control for EIGRP. When enabled, a parent route check is performed in the EIGRP database for controlling PfR prefixes/routes in addition to the existing BGP and static route databases. For more details, see the Using Performance Routing to Control EIGRP Routes with mGRE DMVPN Hub-and-Spoke Support module.

Policy-Based Routing Control

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.

Protocol Independent Route Optimization (PIRO)

In Cisco IOS Release 12.4(24)T, 12.2(33)SRE, and later releases, PIRO was introduced to extend the ability of OER to identify and control traffic classes. Prior to PIRO, OER optimizes paths for traffic classes that have a parent route--an exact matching route, or a less specific route--in BGP or static route databases. PIRO enables OER to search the IP Routing Information Base (RIB) for a parent route allowing OER to be deployed in any IP-routed environment including Interior Gateway Protocols (IGPs) such as OSPF and IS-IS.

The search for a parent route starts in the BGP routing database and, if no parent route is found, the static route database is searched. If a parent route is still not located, the RIB is searched. When a match is found after a parent route search of the RIB, route control is applied to the traffic class using policy-based routing (PBR) where a dynamic route map is created.

After OER route control mode is enabled, no new customer configuration is required to enable PIRO.

On the master controller the **show oer master prefix** command will display PIRO routes as "RIB-PBR" in the output. For more details about debugging PIRO parent route identification and control, see the Verifying and Debugging Protocol Independent Route Optimization Route Control Changes task.

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

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

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.

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.

	Command or Action	Purpose
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.

For an example of setting a BGP local preference value, see the Setting a BGP Local Preference Value for OER Controlled BGP Routes Example.

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*.

Command or Action	Purpose
Step 6 **end** **Example:** `Router(config-oer-mc)# end`	Exits OER master controller configuration mode and returns 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.

The master controller and border routers must be running Cisco IOS Release 12.4(2)T, 12.2(33)SRB, or later releases.

Note

- Border routers must be single-hop peers.
- 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.
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.

Command or Action	Purpose
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*.

	Command or Action	Purpose
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.
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	(Optional) 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 the figure below--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*
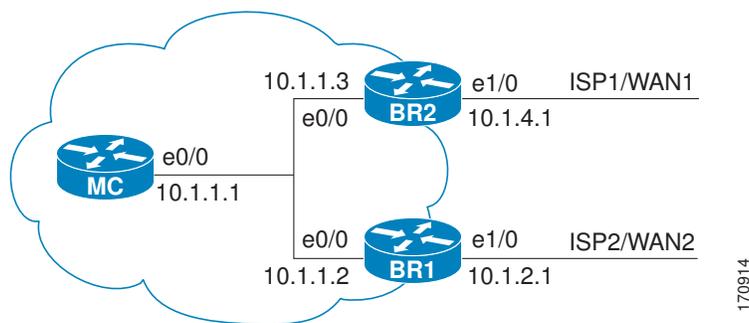
Note　　Policies applied in an OER map do not override global policy configurations.

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.

	Command or Action	Purpose
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*.
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*.

	Command or Action	Purpose
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*.
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.

	Command or Action	Purpose

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.

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.

Command or Action	Purpose
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.
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*.

Command or Action	Purpose
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 two commands are 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.

Example:

`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.

Example:

```
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.

Example:

```
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.

Example:

Step 5   **enable**

Enables privileged EXEC mode. Enter your password if prompted.

Example:

```
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.

Example:

```
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
```

Verifying and Debugging Protocol Independent Route Optimization Route Control Changes

Perform the steps in this optional task if you want to debug PIRO routes where the parent route exists in the RIB and is controlled using policy-based routing. 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 identified using PIRO and that it is being controlled by OER. The first two CLI commands are entered at the master controller, and the other commands are entered at a border router.

This task requires Cisco IOS Release 12.4(24)T, 12.2(33)SRE, or a later release.

SUMMARY STEPS

1. Start at the master controller.
2. **enable**
3. **show oer master traffic-class**
4. Move to a border router to enter the next step.
5. **enable**
6. **show ip route**
7. **show route-map dynamic**
8. **show ip access-list dynamic**
9. **debug oer border routes** {**bgp** | **static**| **piro**[**detail**]}

DETAILED STEPS

Step 1   Start at the master controller.

Step 2   **enable**
Enables privileged EXEC mode. Enter your password if prompted.

Example:

```
Router> enable
```

Step 3   **show oer master traffic-class**
This command is used to display information about traffic classes that are monitored and controlled by an OER master controller. The output from this command includes information about the destination IP address and prefix length for the traffic class, the IP address and the interface of the border router through which the prefix associated with this traffic class is being currently routed, the state of the traffic class, and the protocol. In this example, the protocol displayed for the prefix 10.1.1.0 is RIB-PBR and this means that the parent route for the traffic class exists in the RIB and policy-based routing is being used to control the prefix. Only syntax relevant to this task is shown in this step. You can also use the **show oer master prefix** command to display similar information.

Example:

```
Router# show oer master traffic-class
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

    DstPrefix         Appl_ID Dscp Prot     SrcPort     DstPort SrcPrefix
             Flags               State    Time             CurrBR  CurrI/F Protocol
           PasSDly  PasLDly  PasSUn   PasLUn  PasSLos PasLLos    EBw     IBw
           ActSDly  ActLDly  ActSUn   ActLUn  ActSJit ActPMOS ActSLos  ActLLos
    -------------------------------------------------------------------------------
    10.1.1.0/24            N defa   N            N            N N
                          INPOLICY       0            10.2.1.2   Et4/2  RIB-PBR
                  N       N       N       N       N       N       N       N
                  1       1       0       0       N       N       N       N
```

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.

Example:

```
Router> enable
```

Step 6   **show ip route**

Displays the current state of the routing table. Use this command to verify that a parent route exists in the RIB.

Example:

```
Router# show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, Ethernet4/1
     192.168.0.0/24 is subnetted, 1 subnets
O       192.168.50.0 [110/20] via 10.10.10.3, 00:20:32, Ethernet2/2
     10.0.0.0/8 is variably subnetted, 10 subnets, 4 masks
O       10.1.4.1/32 [110/31] via 10.40.40.2, 00:20:32, Ethernet4/2
O       10.1.5.1/32 [110/31] via 10.40.40.2, 00:20:32, Ethernet4/2
O       10.1.6.1/32 [110/31] via 10.40.40.2, 00:20:32, Ethernet4/2
B       10.1.1.0/24 [20/0] via 10.40.40.2, 00:38:08
     10.1.0.0/24 is subnetted, 1 subnets
O       10.1.1.0 [110/40] via 10.40.40.2, 00:20:33, Ethernet4/2
```

Step 7   **show route-map dynamic**

Viewing a dynamic route map is another method of verifying how the route control is being applied for PIRO routes.
In the output of this dynamic route map, note the access list named oer#6. Only syntax relevant to this task is shown in this step.

Example:

```
Router# show route-map dynamic
```

```
route-map OER-04/21/09-21:42:55.543-6-OER, permit, sequence 0, identifier 1755354068
  Match clauses:
    ip address (access-lists): oer#6
  Set clauses:
    ip next-hop 10.40.40.2
    interface Ethernet4/2
  Policy routing matches: 2314 packets, 138840 bytes
 Current active dynamic routemaps = 1
```

Step 8   **show ip access-list dynamic**

This command displays dynamic IP access lists created on this border router. In the output, a dynamic access list named oer#6, that permits traffic for the prefix 10.1.1.0 to be routed through this border router, is displayed. The access list, oer#6, was identified in the **show route-map dynamic** command in the previous step. Only syntax relevant to this task is shown in this step.

Example:

```
Router# show ip access-list dynamic
Extended IP access list oer#6
    1073741823 permit ip any 10.1.1.0 0.0.0.255 (2243 matches)
```

Step 9   **debug oer border routes** {**bgp** | **static**| **piro**[**detail**]}

This command is entered on a border router. This command is used to debug parent route lookup and route changes to existing parent routes when the parent route is identified from the RIB. In this example, the detailed debugging information shows that the parent route for the prefix 10.1.1.0--shown in the output for Step 2--is found in the RIB and a route map is created to control the application. Note that static and BGP route control, and detailed border PBR debugging is also active.

Example:

```
Router# debug oer border routes piro detail
Apr 21 21:41:25.667: PFR PIRO: Parent lookup found parent 10.1.1.0, mask 24, nexthop
10.40.40.2
Apr 21 21:42:55.539: OER STATIC: No parent found, network 10.1.1.0/24
Apr 21 21:42:55.539: PFR PIRO: Control Route, 10.1.1.0/24, NH 0.0.0.0, IF Ethernet4/2
Apr 21 21:42:55.539: PFR PIRO: Parent lookup found parent 10.1.1.0, mask 24, nexthop
10.40.40.2
Apr 21 21:42:55.539: OER BR PBR(det): control app: 10.1.1.0/24, nh 0.0.0.0, if
Ethernet4/2,ip prot 256, dst opr 0, src opr 0, 0 0 0 0, src net 0.0.0.0/0, dscp 0/0
Apr 21 21:42:55.543: OER BR PBR(det): Create rmap 65DC1CE8
Apr 21 21:42:55.547: PFR PIRO: Parent lookup found parent 10.1.1.0, mask 24, nexthop
10.40.40.2
```

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.

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.

	Command or Action	Purpose
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.
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.

Command or Action	Purpose
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

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 route 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. If a local preference value of 5000 or higher has been configured for default BGP routing, a higher BGP local preference value must be configured in OER. In this task, route control is enabled and a BGP local preference value of 60000 is set.

```
Router(config)# oer master
Router(config-oer-mc)# mode route control
Router(config-oer-mc)# mode route metric bgp local-pref 60000
Router(config-oer-mc)# end
```

ControllingApplicationTraffic 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 task.

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

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS Master Command List	http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html
Command Lookup Tool	http://tools.cisco.com/Support/CLILookup
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
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 and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Using OER to Control Traffic Classes and Verify the Route Control Changes

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

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

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.

Feature Name	Releases	Feature Configuration Information
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 commands were introduced or modified by this feature: **set traceroute reporting**, **traceroute probe-delay**, and **show oer master prefix**.
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 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**.

Feature Name	Releases	Feature Configuration Information
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 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**.
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 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**.

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 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**.
PfR - Protocol Independent Route Optimization (PIRO)	12.2(33)SRE 12.4(24)T	PIRO introduced the ability of OER to search for a parent route-- an exact matching route, or a less specific route--in the IP Routing Information Base (RIB), allowing OER to be deployed in any IP-routed environment including Interior Gateway Protocols (IGPs) such as OSPF and IS-IS._ The following commands were modified by this feature: **debug oer border routes**and **show oer master prefix**.

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