Using OER to Profile the Traffic Classes

First Published: January 29, 2007
Last Updated: August 21, 2007

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.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. 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 63.

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 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

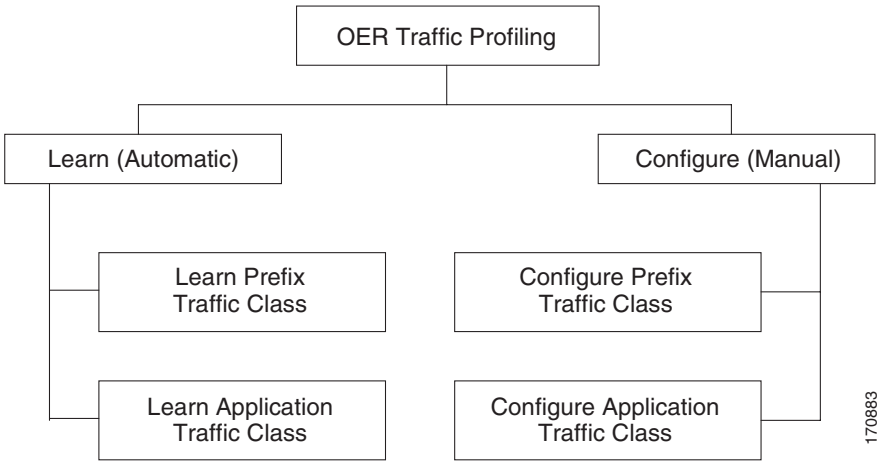
To configure the master controller to profile traffic classes, you should understand the following concepts:

- OER Traffic Class Profiling, page 2
- OER Automatic Traffic Class Learning, page 3
- OER Manual Traffic Class Configuration, page 7

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



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 7

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 3
- 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. A traffic class is automatically learned by OER based on each learn list criteria, and each learn list is configured with a sequence number. The sequence number determines the order in which learn list criteria are applied. Learn lists allow different OER policies to be applied to each learn list; in previous releases, the traffic classes could not be divided, and an OER policy was applied to all the learned traffic classes.

New **traffic-class** commands were introduced under learn list mode to simplify the learning of traffic classes. Three types of traffic classes—to be automatically learned—can be profiled:

- Traffic classes based on destination prefixes
- Traffic classes representing custom application definitions using access lists
- Traffic classes based on a static application mapping name with an optional prefix list filtering to define destination prefixes

Only one type of **traffic-class** command can be specified per learn list, and the **throughput** and **delay** commands are also mutually exclusive within a learn list.

Static Application Mapping Using OER

In Cisco IOS Release 12.4(15)T, the ability to define an application using a keyword was introduced to simplify the configuration of application-based traffic classes. OER uses well-known applications with fixed ports, and more than one application may be configured at the same time. Table 1 displays the

applications that can be configured. The applications are considered static because they are defined with fixed port and protocols as shown in the table. Configuration is performed on a master controller under learn list configuration mode.

Table 1 Static Application List

Application	Protocol	Port
CU-SeeMe-Server—CU-SeeMe desktop video conference	TCP UDP	7648 7649 7648 7649 24032
DHCP-Client—Dynamic Host Configuration Protocol client	UDP/TCP	68
DHCP-Server—Dynamic Host Configuration Protocol server	UDP/TCP	67
DNS—Domain Name Server lookup	UDP/TCP	53
FINGER-Server—Finger server	TCP	79
GOPHER-Server—Gopher server	TCP/UDP	70
HTTP—Hypertext Transfer Protocol, World Wide Web traffic	TCP/UDP	80
HTTPSSL-Server—Hypertext Transfer Protocol over TLS/SSL, Secure World Wide Web traffic server	TCP	443
IMAP-Server—Internet Message Access Protocol server	TCP/UDP	143 220
SIMAP-Server—Secure Internet Message Access Protocol server	TCP/UDP	585 993 (Preferred)
IRC-Server—Internet Relay Chat server	TCP/UDP	194
SIRC-Server—Secure Internet Relay Chat server	TCP/UDP	994
Kerberos-Server—Kerberos server	TCP/UDP	88 749
L2TP-Server—L2F/L2TP tunnel Layer 2 Tunnel Protocol server	UDP	1701
LDAP-Server—Lightweight Directory Access Protocol server	TCP/UDP	389
SLDAP-Server—Secure Lightweight Directory Access Protocol server	TCP/UDP	636
MSSQL-Server—MS SQL server	TCP	1443
NETBIOS-Server—NETBIOS server	UDP TCP	137 138 137 139
NFS-Server—Network File System server	TCP/UDP	2049
NNTP-Server—Network News Transfer Protocol	TCP/UDP	119
SNNTP-Server—Network News Transfer Protocol over TLS/SSL	TCP/UDP	563
NOTES-Server—Lotus Notes server	TCP/UDP	1352
NTP-Server—Network Time Protocol server	TCP/UDP	123
PCanywhere-Server—Symantec pcANYWHERE	UDP TCP	22 5632 65301 5631
POP3-Server—Post Office Protocol server	TCP/UDP	110
SPOP3-Server—Post Office Protocol over TLS/SSL server	TCP/UDP	123

Table 1 Static Application List (continued)

Application	Protocol	Port
PPTP-Server—Point-to-Point Tunneling Protocol server	TCP	17233
SMTP-Server—Simple Mail Transfer Protocol server	TCP	25

The master controller is configured to learn the top prefixes based on highest outbound throughput or delay for the filtered traffic, and the resulting traffic classes are added to the OER application database to be passively and actively monitored.

For more details on configuring application-based traffic classes using static application mapping, see the "Defining a Learn List to Automatically Learn Traffic Classes Using Static Application Mapping" section on page 16.

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 7
- Application Traffic Class Configuration Using OER, page 8

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 **match traffic-class** commands were introduced under OER map configuration mode to simplify the configuration of traffic classes. Three types of traffic classes—to be manually configured—can be profiled:

- Traffic classes based on destination prefixes
- Traffic classes representing custom application definitions using access lists
- Traffic classes based on a static application mapping name with an optional prefix list filtering to define destination prefixes

Only one type of **match traffic-class** command can be specified per OER map.

For a series of well-known applications, static ports have been defined and each application can be defined by entering a keyword. For more details about the mapping of keywords to static applications using OER, see the "Static Application Mapping Using OER" section on page 5.

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. 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. New commands were also introduced to simplify the profiling of traffic classes. This section is divided into two task groups to show the distinction between the new application aware routing configuration methods of profiling the traffic classes and the existing configuration methods of profiling the traffic classes:

Configuring Application Aware Routing to Profile the Traffic Classes in Cisco IOS Release 12.4(15)T and Later Releases

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. The ability to learn application traffic classes using a keyword representing a static application is also introduced and new **traffic-class** and **match traffic-class** commands simplify the configuration of traffic classes that OER can automatically learn, or that can be manually configured.

Note 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.

Perform any of the following tasks to profile traffic classes:

Defining a Learn List for Automatically Learned Prefix-Based Traffic Classes

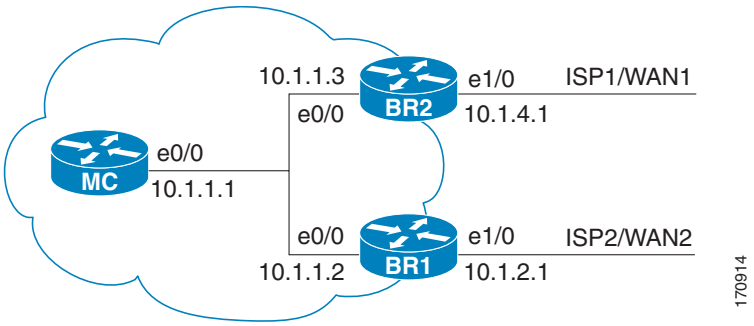
Perform this task at the master controller to define a learn list that will contain traffic classes that are automatically learned based on a prefix list. In Cisco IOS Release 12.4(15)T learn lists were introduced to allow traffic classes to be categorized. Learn lists allow different OER policies to be applied to each learn list; in previous releases the traffic classes could not be divided, and an OER policy was applied to all the learned traffic classes.

This task is performed on the master controller shown in Figure 3. In this task, a prefix list is created to specify destination addresses and a learn list is defined under OER Top Talker and Top Delay configuration mode. This task configures prefix learning based on the highest outbound throughput.

In this task, the IP prefix list specifies that the prefix 10.1.0.0/16 is to be used as filter when learning the traffic classes. Under learn list mode, the prefix length aggregation is configured as a /24 prefix length. The traffic classes learned by this task will be:

```
10.1.1.0/24
10.1.2.0/24
.
.
.
10.1.255.0/24
```

Figure 2 Network Diagram of OER Master Controller and Border Routers



Prerequisites

This task requires the master controller and border routers to be running Cisco IOS Release 12.4(15)T or a later release.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip prefix-list**-*name* [**seq** *seq-value*] {**deny** *network/length* | **permit** *network/length*} [**le** *le-value*]
4. Repeat Step 3 for more prefix list entries, as required.
5. **oer master**
6. **learn**
7. **list seq** *number* **refname** *refname*
8. **count** *number* **max** *max-number*
9. **traffic-class prefix-list** *prefix-list-name*

10. **aggregation-type** {**bgp** | **non-bgp** | **prefix-length** *prefix-mask*}
11. **throughput**
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	**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 10 permit 10.1.0.0/16`	Creates an IP prefix list. • An IP prefix list is used under learn list configuration mode to filter IP addresses that are learned. • The example creates an IP prefix list named PREFIXES for OER to profile the prefix, 10.1.0.0/16.
Step 4	Repeat Step 3 for more prefix list entries, as required.	—
Step 5	**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 6	**learn** **Example:** `Router(config-oer-mc)# learn`	Enters OER Top Talker and Top Delay learning configuration mode to automatically learn traffic classes.
Step 7	**list seq** *number* **refname** *refname* **Example:** `Router(config-oer-mc-learn)# list seq 10 refname LEARN_PREFIXES_TC`	Creates an OER learn list and enters learn list configuration mode. • Use the **seq** keyword and *number* argument to specify a sequence number used to determine the order in which learn list criteria is applied. • Use the **refname** keyword and *refname* argument to specify a reference name for the learn list. • The example creates a learn list named LEARN_PREFIXES_TC.

	Command or Action	Purpose
Step 8	**count** *number* **max** *max-number* **Example:** `Router(config-oer-mc-learn-list)# count 40 max 90`	Sets the number of traffic classes to be learned during an OER learn session. • Use the *number* argument to specify a number of traffic classes to be learned for the specified learn list during a learn session. • Use the **max** keyword and *max-number* argument to specify a maximum number of traffic classes to be learned for the specified learn list during all learning sessions. • The example specifies 40 traffic classes to be learned per learning session for the learn list named LEARN_PREFIXES_TC, and a maximum of 90 traffic classes in total for this learn list.
Step 9	**traffic-class prefix-list** *prefix-list-name* [**inside**] **Example:** `Router(config-oer-mc-learn-list)# traffic-class prefix-list LEARN_PREFIXES_TC`	Defines an OER traffic class using a prefix list. • Use the *prefix-list-name* argument to specify an IP prefix list that contains criteria for defining the traffic classes. • Use the **inside** keyword to specify inside prefixes. • The example uses the prefix list named LEARN_PREFIXES_TC to filter the learned traffic classes.
Step 10	**aggregation-type** {**bgp** | **non-bgp** | **prefix-length**} *prefix-mask* **Example:** `Router(config-oer-mc-learn-list)# 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 based on a /24 prefix length.

	Command or Action	Purpose
Step 11	**throughput** **Example:** `Router(config-oer-mc-learn-list)# throughput`	Enables prefix learning based on the highest outbound throughput. • Prefixes are sorted from the highest to lowest outbound throughput. • The example configures prefix learning based on the highest outbound throughput. **Note** To configure automatic OER learning within a learn list you can specify either the **delay** command or the **throughput** command, but they are mutually exclusive in learn list configuration mode.
Step 12	**end** **Example:** `Router(config-oer-mc-learn-list)# end`	Exits learn list configuration mode, and returns to privileged EXEC mode.

Defining a Learn List for Automatically Learned Application Traffic Classes Using an Access List

Perform this task at the master controller to define a learn list that will contain traffic classes that are automatically learned by OER using an access list to create customized application traffic classes. Use this task to build application traffic classes where the traffic cannot be defined using standard application port and protocol mapping. If there are some known prefixes that you want to exclude, an optional prefix list can be used to further filter the traffic although this is not shown in this task.

In Cisco IOS Release 12.4(15)T learn lists were introduced to allow traffic classes to be categorized. Learn lists allow different OER policies to be applied to each learn list; in previous releases the traffic classes could not be divided, and an OER policy was applied to all the traffic classes profiled during one learning session.

In this task, an access list is created that defines custom application traffic classes. Every entry in the access list defines one application. A learn list is then defined, the access list is applied, and an aggregation method is configured. Using the **count** command, 50 traffic classes can be learned during one learning session for the learn list named LEARN_USER_DEFINED_TC, with a maximum specified number of 90 traffic classes for this learn list. The master controller is configured to learn the top prefixes based on highest delay for the filtered traffic and the resulting traffic classes are added to the OER application database.

To display information about the traffic classes learned by OER, use the "Displaying and Resetting Traffic Class and Learn List Information" section on page 25.

Prerequisites

This task requires the master controller and border routers to be running Cisco IOS Release 12.4(15)T or a 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*]] [**dscp** *dscp-value*]
5. Repeat Step 4 for more access list entries, as required.
6. **exit**
7. **oer master**
8. **learn**
9. **list seq** *number* **refname** *refname*
10. **count** *number* **max** *max-number*
11. **traffic-class access-list** *access-list-name* [**filter** *prefix-list-name*]
12. **aggregation-type** {**bgp** | **non-bgp** | **prefix-length** *prefix-mask*}
13. **delay**
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.
Step 3	**ip access-list** {**standard** | **extended**} *access-list-name* **Example:** `Router(config)# ip access-list extended USER_DEFINED_TC`	Defines an IP access list by name. • OER supports only named access lists. • The example creates an extended IP access list named USER_DEFINED_TC.
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 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 5	Repeat Step 4 for more access list entries, as required.	—
Step 6	**exit** **Example:** `Router(config-ext-nacl)# exit`	(Optional) Exits extended access list configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 7	**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 8	**learn** **Example:** `Router(config-oer-mc)# learn`	Enters OER Top Talker and Top Delay learning configuration mode to automatically learn traffic classes.
Step 9	**list seq** *number* **refname** *refname* **Example:** `Router(config-oer-mc-learn)# list seq 10 refname LEARN_USER_DEFINED_TC`	Creates an OER learn list and enters learn list configuration mode. • Use the **seq** keyword and *number* argument to specify a sequence number used to determine the order in which learn list criteria is applied. • Use the **refname** keyword and *refname* argument to specify a reference name for the learn list. • The example creates a learn list named LEARN_USER_DEFINED_TC.
Step 10	**count** *number* **max** *max-number* **Example:** `Router(config-oer-mc-learn-list)# count 50 max 90`	Sets the number of traffic classes to be learned during an OER learn session. • Use the *number* argument to specify a number of traffic classes to be learned for the specified learn list during a learn session. • Use the **max** keyword and *max-number* argument to specify a maximum number of traffic classes to be learned for the specified learn list during all learning sessions. • The example specifies 50 traffic classes to be learned per learning session for the learn list named LEARN_USER_DEFINED_TC, and a maximum of 90 traffic classes in total for this learn list.
Step 11	**traffic-class access-list** *access-list-name* [**filter** *prefix-list-name*] **Example:** `Router(config-oer-mc-learn-list)# traffic-class access-list USER_DEFINED_TC`	Defines an OER traffic class using an access list. • Use the *access-list-name* argument to specify an access list that contains criteria for defining the traffic classes. • The example uses the access list named USER_DEFINED_TC to create the traffic classes.

	Command or Action	Purpose
Step 12	**aggregation-type** {**bgp** | **non-bgp** | **prefix-length**} *prefix-mask* **Example:** `Router(config-oer-mc-learn-list)# 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 based on a /24 prefix length.
Step 13	**delay** **Example:** `Router(config-oer-mc-learn-list)# 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 automatic OER learning within a learn list you can specify either the **delay** command or the **throughput** command, but they are mutually exclusive in learn list configuration mode.
Step 14	**end** **Example:** `Router(config-oer-mc-learn-list)# end`	Exits learn list configuration mode, and returns to privileged EXEC mode.

Defining a Learn List to Automatically Learn Traffic Classes Using Static Application Mapping

Perform this task at the master controller to define a learn list using static application mapping. Within a learn list, a keyword that represents an application can be used to identify specific application traffic classes. The defined learn list will contain traffic classes to be automatically learned by OER using the static application mapping. The resulting traffic classes can be filtered by a prefix list, if required.

In Cisco IOS Release 12.4(15)T learn lists were introduced to allow traffic classes to be categorized. Learn lists allow different OER policies to be applied to each learn list; in previous releases the traffic classes could not be divided, and an OER policy was applied to all the traffic classes profiled during one learning session.

In this example, two learn lists are configured to identify remote login traffic and file transfer traffic. The remote login traffic class is configured using keywords representing Telnet and Secure Shell (SSH) traffic, and the resulting prefixes are aggregated to a prefix length of 24. The file transfer traffic class is configured using a keyword that represents FTP and is also aggregated to a prefix length of 24. A prefix

list is applied to the file transfer traffic class to permit traffic from the 10.0.0.0/8 prefix. The master controller is 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 display information about the configured learn lists and the traffic classes learned by OER, use the "Displaying and Resetting Traffic Class and Learn List Information" section on page 25.

Prerequisites

This task requires the master controller and border routers to be running Cisco IOS Release 12.4(15)T or a later release.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip prefix-list***-name* [**seq** *seq-value*] {**deny** *network/length* | **permit** *network/length*} [**le** *le-value*]
4. **oer master**
5. **learn**
6. **list seq** *number* **refname** *refname*
7. **traffic-class application** *application-name* [**filter** *prefix-list-name*]
8. **aggregation-type** {**bgp** | **non-bgp** | **prefix-length** *prefix-mask*}
9. **throughput**
10. **exit**
11. **list seq** *number* **refname** *refname*
12. **traffic-class application** *application-name* [**filter** *prefix-list-name*]
13. **aggregation-type** {**bgp** | **non-bgp** | **prefix-length** *prefix-mask*}
14. **throughput**
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.

	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 INCLUDE_10_NET permit 10.0.0.0/8`	Creates an IP prefix list to filter prefixes for learning. • An IP prefix list is used under learn list configuration mode to filter IP addresses that are learned. • The example creates an IP prefix list named INCLUDE_10_NET for OER to profile the prefix, 10.0.0.0/8.
Step 4	**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 5	**learn** **Example:** `Router(config-oer-mc)# learn`	Enters OER Top Talker and Top Delay learning configuration mode to automatically learn traffic classes.
Step 6	**list seq** *number* **refname** *refname* **Example:** `Router(config-oer-mc-learn)# list seq 10 refname LEARN_REMOTE_LOGIN_TC`	Creates an OER learn list and enters learn list configuration mode. • Use the **seq** keyword and *number* argument to specify a sequence number used to determine the order in which learn list criteria is applied. • Use the **refname** keyword and *refname* argument to specify a reference name for the learn list. • The example creates a learn list named LEARN_REMOTE_LOGIN_TC.
Step 7	**traffic-class application** *application-name...* [**filter** *prefix-list-name*] **Example:** `Router(config-oer-mc-learn-list)# traffic-class application telnet ssh`	Defines an OER traffic class using a pre-defined static application. • Use the *application-name* argument to specify one or more keywords that represent pre-defined static applications. The ellipses are used to show that more than one application keyword can be specified. • The example defines a traffic class as containing telnet and ssh traffic.

	Command or Action	Purpose
Step 8	**aggregation-type** {**bgp** | **non-bgp** | **prefix-length**} *prefix-mask* **Example:** `Router(config-oer-mc-learn-list)# 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 based on a /24 prefix length.
Step 9	**throughput** **Example:** `Router(config-oer-mc-learn-list)# 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 for the LEARN_REMOTE_LOGIN_TC traffic class.
Step 10	**exit** **Example:** `Router(config-oer-mc-learn-list)# exit`	Exits learn list configuration mode, and returns to OER Top Talker and Top Delay learning configuration mode.
Step 11	**list seq** *number* **refname** *refname* **Example:** `Router(config-oer-mc-learn)# list seq 20 refname LEARN_FILE_TRANSFER_TC`	Creates an OER learn list and enters learn list configuration mode. • Use the **seq** keyword and *number* argument to specify a sequence number used to determine the order in which learn list criteria is applied. • Use the **refname** keyword and *refname* argument to specify a reference name for the learn list. • The example creates a learn list named LEARN_FILE_TRANSFER_TC.

	Command or Action	Purpose
Step 12	**traffic-class application** *application-name...* [**filter** *prefix-list-name*] **Example:** `Router(config-oer-mc-learn-list)# traffic-class application ftp filter INCLUDE_10_NET`	Defines an OER traffic class using a pre-defined static application. • Use the *application-name* argument to specify one or more keywords that represent pre-defined static applications. • The example defines a traffic class as containing FTP traffic filtered by the IP prefix list named INCLUDE_10_NET to include prefixes in the 10.0.0.0/8 subnet.
Step 13	**aggregation-type** {**bgp** | **non-bgp** | **prefix-length**} *prefix-mask* **Example:** `Router(config-oer-mc-learn-list)# 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 based on a /24 prefix length.
Step 14	**throughput** **Example:** `Router(config-oer-mc-learn-list)# 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 for the LEARN_FILE_TRANSFER_TC traffic class. **Note** To configure automatic OER learning within a learn list you can specify either the **delay** command or the **throughput** command, but they are mutually exclusive in learn list configuration mode.
Step 15	**end** **Example:** `Router(config-oer-mc-learn-list)# end`	Exits learn list configuration mode, and returns to privileged EXEC mode.

Manually Selecting Traffic Classes Using Static Application Mapping

Perform this task to manually select traffic classes using static application mapping. Use this task when you know the destination prefixes and the applications that you want to select for the traffic classes. In this task, an IP prefix list is created to define the destination prefixes, and static applications are defined using the **match traffic-class application** command. Using an OER map, each prefix is matched with each application to create the traffic classes.

Prerequisites

This task requires the master controller and border routers to be running Cisco IOS Release 12.4(15)T, or a later release.

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. Repeat Step 3 for more prefix list entries, as required.
5. **oer-map** *map-name sequence-number*
6. **match traffic-class application** *application-name* **prefix-list** *prefix-list-name*
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 LIST1 permit 10.1.1.0/24`	Creates a prefix list to specify destination prefix-based traffic classes. • The example specifies a destination prefix of 10.1.1.0/24 to be used to filter application traffic classes.
Step 4	Repeat Step 3 for more prefix list entries, as required.	—

	Command or Action	Purpose
Step 5	**oer-map** *map-name sequence-number* **Example:** `Router(config)# oer-map APPLICATION_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 6. • The example creates an OER map named APPLICATION_MAP.
Step 6	**match traffic-class application** *application-name* **prefix-list** *prefix-list-name* **Example:** `Router(config-oer-map)# traffic-class application telnet ssh filter LIST1`	Manually configures one or more static applications as match criteria against a prefix list to create traffic classes using an OER map. • Use the *application-name* argument to specify one or more keywords that represent pre-defined static applications. • The example defines traffic classes as application X with destination prefix Y, where X is Telnet or Secure Shell and Y is a destination address defined in the IP prefix list named LIST1.
Step 7	**end** **Example:** `Router(config-oer-map)# end`	(Optional) Exits OER map configuration mode and returns to privileged EXEC mode.

Manually Selecting Prefix-Based Traffic Classes Using a Prefix List

Perform this task on the master controller to manually select traffic classes based only on destination prefixes. Use this task when you know the destination prefixes that you want to select for the traffic classes. An IP prefix list is created to define the destination prefixes and using an OER map, the traffic classes are profiled.

Prerequisites

This task requires the master controller and border routers to be running Cisco IOS Release 12.4(15)T or a later release.

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. Repeat Step 3 for more prefix list entries, as required.
5. **oer-map** *map-name sequence-number*
6. **match traffic-class prefix-list** *prefix-list-name*
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 PREFIX_TC permit 172.16.1.0/24`	Creates a prefix list to specify destination prefix-based traffic classes. • The example creates a prefix list named PREFIX_TC that specifies a destination prefix of 172.16.1.0/24 to be selected for a traffic class.
Step 4	Repeat Step 3 for more prefix list entries, as required.	—
Step 5	**oer-map** *map-name sequence-number* **Example:** `Router(config)# oer-map PREFIX_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 6. • The example creates an OER map named PREFIX_MAP.
Step 6	**match traffic-class prefix-list** *prefix-list-name* **Example:** `Router(config-oer-map)# match traffic-class prefix-list PREFIX_TC`	Manually configures a prefix list as match criteria used to create traffic classes using an OER map. • The example defines a traffic class using the destination address defined in the IP prefix list named PREFIX_TC.
Step 7	**end** **Example:** `Router(config-oer-map)# end`	(Optional) Exits OER map configuration mode and returns to privileged EXEC mode.

Manually Selecting Application Traffic Classes Using an Access List

Perform this task on the master controller to manually select traffic classes using an access list. Each access list entry is a traffic class that must include a destination prefix and may include other optional parameters. This task uses the **match traffic-class access-list** command, which is the similar to using the **match ip-address** command in previous releases. In this task, an access list is created and using an OER map, the traffic classes are profiled.

Prerequisites

This task requires the master controller and border routers to be running Cisco IOS Release 12.4(15)T or a 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*]] [**dscp** *dscp-value*]
5. Repeat Step 4 for more access list entries, as required.
6. **exit**
7. **oer-map** *map-name sequence-number*
8. **match traffic-class access-list** *access-list-name*
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 access-list** {**standard** | **extended**} *access-list-name* **Example:** `Router(config)# ip access-list extended ACCESS_TC`	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_TC.
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 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 5	Repeat Step 4 for more access list entries, as required.	—

	Command or Action	Purpose
Step 6	**exit** **Example:** `Router(config-ext-nacl)# exit`	(Optional) Exits extended named access list configuration mode and returns to global configuration mode.
Step 7	**oer-map** *map-name sequence-number* **Example:** `Router(config)# oer-map ACCESS_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 6. • The example creates an OER map named ACCESS_MAP.
Step 8	**match traffic-class access-list** *access-list-name* **Example:** `Router(config-oer-map)# match traffic-class access-list ACCESS_TC`	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_TC.
Step 9	**end** **Example:** `Router(config-oer-map)# end`	(Optional) Exits OER map configuration mode and returns to privileged EXEC mode.

Displaying and Resetting Traffic Class and Learn List Information

Perform this task to display traffic class and learn list information and optionally, to reset some traffic class information. These commands can be entered on a master controller after learn lists are configured and traffic classes are automatically learned, or when traffic classes are manually configured using an OER map. The commands can be entered in any order and all the commands are optional.

Prerequisites

This task requires the master controller to be running Cisco IOS Release 12.4(15)T.

SUMMARY STEPS

1. **enable**
2. **show oer master traffic-class** [**access-list** *access-list-name* | **application** *application-name* [*prefix*] | **inside** | **learned** [**delay** | **inside** | **list** *list-name* | **throughput**] | **prefix** *prefix* | **prefix-list** *prefix-list-name*] [**active** | **passive** | **status**] [**detail**]
3. **show oer master learn list** *list-name*
4. **clear oer master traffic-class** [**access-list** *access-list-name* | **application** *application-name* [*prefix*] | **inside** | **learned** [**delay** | **inside** | **list** *list-name* | **throughput**] | **prefix** *prefix* | **prefix-list** *prefix-list-name*]

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

```
Router> enable
```

Step 2 **show oer master traffic-class** [**access-list** *access-list-name* | **application** *application-name* [*prefix*] | **inside** | **learned** [**delay** | **inside** | **list** *list-name* | **throughput**] | **prefix** *prefix* | **prefix-list** *prefix-list-name*] [**active** | **passive** | **status**] [**detail**]

This command is used to display information about traffic classes learned or manually configured under OER learn list configuration mode.

```
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
--------------------------------------------------------------------------------
10.1.1.0/24              N defa    N           N           N N
               #          OOPOLICY       32         10.11.1.3    Et1/0      BGP
               N        N        N        N        N        N        N      IBwN
             130      134        0        0        N        N
```

Step 3 **show oer master learn list** [*list-name*]

This command is used to display one or all of the configured OER learn lists. In this example, the information about two learn lists is displayed.

```
Router# show oer master learn list

Learn-List LIST1 10
   Configuration:
    Application: ftp
    Aggregation-type: bgp
    Learn type: thruput
    Policies assigned: 8 10
   Stats:
    Application Count: 0
    Application Learned:
 Learn-List LIST2 20
   Configuration:
    Application: telnet
    Aggregation-type: prefix-length 24
    Learn type: thruput
    Policies assigned: 5 20
   Stats:
    Application Count: 2
    Application Learned:
     Appl Prefix 10.1.5.0/24 telnet
     Appl Prefix 10.1.5.16/28 telnet
```

Step 4 **clear oer master traffic-class** [**access-list** *access-list-name* | **application** *application-name* [*prefix*] | **inside** | **learned** [**delay** | **inside** | **list** *list-name* | **throughput**] | **prefix** *prefix* | **prefix-list** *prefix-list-name*]

This command is used to clear OER controlled traffic classes from the master controller database. The following example clears traffic classes defined by the Telnet application and the 10.1.1.0/24 prefix:

```
Router# clear oer master traffic-class application telnet 10.1.1.0/24
```

Configuring OER to Profile the Traffic Classes in Cisco IOS Release 12.4(11)T, 12.2(33)SRB, or Earlier Releases

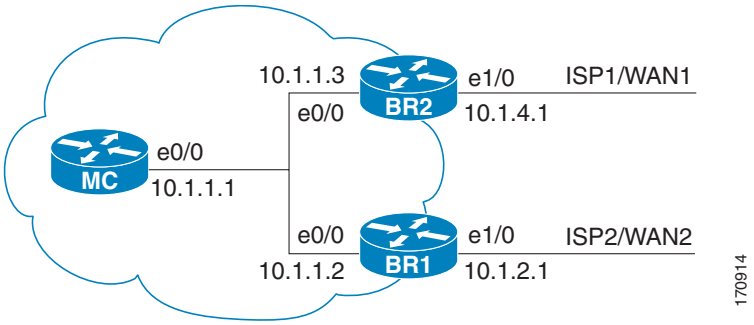
In Cisco IOS Release 12.4(11)T, Release 12.2(33)SRB, and earlier releases, an OER master controller can be configured to automatically learn the traffic classes, or the traffic classes can be manually configured. Protocols, port numbers, and DSCP bits can be specified in addition to prefixes. One or more of the following tasks may be performed:

- Configuring OER to Automatically Learn Prefix-Based Traffic Classes, page 27
- Configuring OER to Automatically Learn Traffic Classes Using Inside Prefixes, page 31
- Configuring OER to Automatically Learn Prefix-Based Traffic Classes Using Protocol or Port Number, page 33
- Specifying the Flow Keys for Automatic Learning of Application Traffic Classes, page 36
- Creating an Access List to Specify a Filter for Automatically Learned Application Traffic, page 39
- Creating an Access List to Specify Aggregation Criteria for Automatically Learned Application Traffic, page 43
- Displaying Application Traffic Flow Information on a Border Router, page 47
- Manually Selecting Prefixes for OER Monitoring, page 49
- Manually Selecting Inside Prefixes for OER Monitoring, page 51
- Manually Selecting Traffic Classes Using Prefix, Protocol, Port, and DSCP Value, page 53

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 3.

Figure 3 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.

	Command or Action	Purpose
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.
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.

	Command or Action	Purpose
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.
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 49.

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.

	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 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.
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.

	Command or Action	Purpose
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 51.

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.
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. • 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.
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	**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 53.

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 47.

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 47.

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*, 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 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 47.

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*, 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 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, 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 7.

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 27.

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 7.

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 31.

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

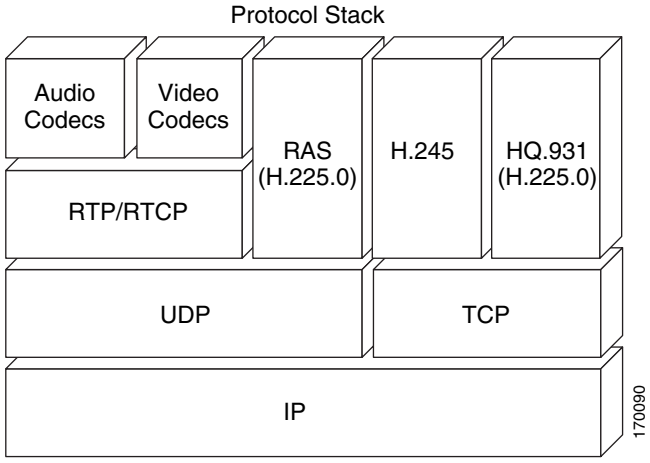
Perform this task to manually select traffic classes using prefixes, protocols, port numbers, and DSCP value for OER monitoring. An IP access list is created to define the parameters to identify the traffic classes. The access list can then be imported into the MTC list by configuring a match clause 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 4 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 4 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 36.

Configuration Examples for Using OER to Profile the Traffic Classes

This section is divided into two configuration example groups to show the distinction between the new application aware routing configuration methods of profiling the traffic classes introduced in Cisco IOS Release 12.4(15)T and later releases, and the existing configuration methods of profiling the traffic classes:

- Configuring Application Aware Routing to Profile the Traffic Classes in Cisco IOS Release 12.4(15)T and Later Releases: Examples, page 56
- Configuring OER to Profile the Traffic Classes in Cisco IOS Release 12.4(11)T, 12.2(33)SRB, or Earlier Releases: Examples, page 59

Configuring Application Aware Routing to Profile the Traffic Classes in Cisco IOS Release 12.4(15)T and Later Releases: Examples

The configuration examples in this section show how to use application aware routing techniques introduced in Cisco IOS Release 12.4(15)T and later releases, to profile the traffic classes:

Note 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.

- Defining a Learn List for Automatically Learned Prefix-Based Traffic Classes: Example, page 57
- Defining a Learn List for Automatically Learned Application Traffic Classes Using an Access List: Example, page 57
- Defining a Learn List to Automatically Learn Traffic Classes Using Static Application Mapping: Example, page 58
- Manually Selecting Traffic Classes Using Static Application Mapping: Example, page 58
- Manually Selecting Prefix-Based Traffic Classes Using a Prefix List: Example, page 59
- Manually Selecting Application Traffic Classes Using an Access List: Example, page 59

Defining a Learn List for Automatically Learned Prefix-Based Traffic Classes: Example

The following example configured on the master controller, defines a learn list that will contain traffic classes that are automatically learned based only on a prefix list. In this example, there are three branch offices and the goal is to optimize all the traffic going to branch offices A and B using one policy (Policy1), and to optimize traffic going to branch office C using a different policy (Policy2).

Branch A is defined as any prefix that matches 10.1.0.0./16, Branch B is defined as any prefix that matches 10.2.0.0./16, and Branch C is defined as any prefix that matches 10.3.0.0./16.

This task configures prefix learning based on the highest outbound throughput.

```
ip prefix-list BRANCH_A_B permit seq 10 10.1.0.0/16
ip prefix-list BRANCH_A_B permit seq 20 10.2.0.0/16
ip prefix-list BRANCH_C permit seq 30 10.3.0.0/16
oer master
 learn
 list seq 10 refname LEARN_BRANCH_A_B
 traffic-class prefix-list BRANCH_A_B
 throughput
 exit
 exit
 learn
 list seq 20 refname LEARN_BRANCH_C
 traffic-class prefix-list BRANCH_C
 throughput
 exit
 exit
oer-map POLICY1 10
 match learn list LEARN_BRANCH_A_B
 exit
oer-map POLICY2 10
 match learn list LEARN_BRANCH_C
 end
```

Defining a Learn List for Automatically Learned Application Traffic Classes Using an Access List: Example

The following example creates an access list that defines custom application traffic classes. In this example, the custom application consists of four criteria:

- Any TCP traffic on destination port 500
- Any TCP traffic on ports in the range from 700 to 750
- Any UDP traffic on source port 400
- Any IP packet marked with a DSCP bit of ef

The goal is to optimize same policy POLICY_CUSTOM_APP is to be applied to all the learned traffic classes. This task configures traffic class learning based on the highest outbound throughput.

```
ip access-list extended USER_DEFINED_TC
 permit tcp any any 500
 permit tcp any any range 700 750
 permit udp any eq 400 any
 permit ip any any dscp ef
 exit
oer master
 learn
 list seq 10 refname CUSTOM_APPLICATION_TC
 traffic-class access-list USER_DEFINED_TC
 aggregation-type prefix-length 24
```

```
 throughput
 exit
 exit
oer-map POLICY_CUSTOM_APP 10
 match learn list CUSTOM_APPLICATION_TC
 end
```

Defining a Learn List to Automatically Learn Traffic Classes Using Static Application Mapping: Example

The following example defines application traffic classes using static application mapping. In this example, the following two OER learn lists are defined:

- LEARN_REMOTE_LOGIN_TC—Remote login traffic represented by Telnet and SSH.
- LEARN_FILE_TRANSFER_TC—File transfer traffic represented by FTP and filtered by the 10.0.0.0/8 prefix.

The goal is to optimize the remote login traffic using one policy (POLICY_REMOTE), and to optimize the file transfer traffic using a different policy (POLICY_FILE). This task configures traffic class learning based on the highest delay.

```
ip prefix-list INCLUDE_10_NET 10.0.0.0/8
oer master
 learn
 list seq 10 refname LEARN_REMOTE_LOGIN_TC
 traffic-class application telnet ssh
 aggregation-type prefix-length 24
 delay
 exit
 list seq 20 refname LEARN_FILE_TRANSFER_TC
 traffic-class application ftp filter INCLUDE_10_NET
 aggregation-type prefix-length 24
 delay
 exit
 exit
oer-map POLICY_REMOTE 10
 match learn list LEARN_REMOTE_LOGIN_TC
 exit
oer-map POLICY_FILE 20
 match learn list LEARN_FILE_TRANSFER_TC
 end
```

Manually Selecting Traffic Classes Using Static Application Mapping: Example

The following example starting in global configuration mode, configures an OER map to include application traffic predefined as telnet or Secure Shell and destined to prefixes in the 10.1.1.0/24 network, 10.1.2.0/24 network, and 172.16.1.0/24 network.

```
ip prefix-list LIST1 permit 10.1.1.0/24
ip prefix-list LIST1 permit 10.1.2.0/24
ip prefix-list LIST1 permit 172.16.1.0/24
oer-map PREFIXES 10
 match traffic-class application telnet ssh prefix-list LIST1
 end
```

Manually Selecting Prefix-Based Traffic Classes Using a Prefix List: Example

The following example configured on the master controller, manually selects traffic classes based only on destination prefixes. Use this task when you know the destination prefixes that you want to select for the traffic classes. An IP prefix list is created to define the destination prefixes and using an OER map, the traffic classes are profiled.

```
ip prefix-list PREFIX_TC permit 10.1.1.0/24
ip prefix-list PREFIX_TC permit 10.1.2.0/24
ip prefix-list PREFIX_TC permit 172.16.1.0/24
oer-map PREFIX_MAP 10
 match traffic-class prefix-list PREFIX_TC
```

Manually Selecting Application Traffic Classes Using an Access List: Example

The following example configured on the master controller, manually selects traffic classes using an access list. Each access list entry is a traffic class that must include a destination prefix and may include other optional parameters.

```
ip access-list extended ACCESS_TC
 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 800any any dscp ef
 exit
oer-map ACCESS_MAP 10
 match traffic-class access-list ACCESS_TC
```

Configuring OER to Profile the Traffic Classes in Cisco IOS Release 12.4(11)T, 12.2(33)SRB, or Earlier Releases: Examples

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

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:

- 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 the 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 63.

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
Cisco OER commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	• *Cisco IOS Optimized Edge Routing Command Reference*, Release 12.4T • *Cisco IOS Optimized Edge Routing Command Reference*, Release 12.2 SR
IP prefix list commands	• *Cisco IOS IP Routing Protocols Command Reference*, Release 12.4T • *Cisco IOS IP Routing Protocols Command Reference*, Release 12.2SR
IP prefix list configuration tasks	*Cisco IOS IP Routing Protocols Configuration Guide*, Release 12.4

Technical Assistance

Description	Link
The Cisco Technical Support & Documentation website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, tools, and technical documentation. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Feature Information for Using OER to Profile the Traffic Classes

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(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 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 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 3 • Prefix Traffic Class Configuration Using OER, page 7 • Configuring OER to Automatically Learn Prefix-Based Traffic Classes Using Protocol or Port Number, page 33 • Manually Selecting Traffic Classes Using Prefix, Protocol, Port, and DSCP Value, page 53 • Configuring OER to Automatically Learn Traffic Classes Using Prefixes and Protocol or Port Numbers: Example, page 60 • Manually Selecting Traffic Classes Using Prefix, Protocol, Port, and DSCP Value: Example, page 62 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).

Table 2 Feature Information for Using OER to Profile the Traffic Classes (continued)

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: • Application Traffic Class Configuration Using OER, page 8 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 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 3 • Configuring OER to Automatically Learn Traffic Classes Using Inside Prefixes, page 31 • Manually Selecting Inside Prefixes for OER Monitoring, page 51 • Configuring OER to Automatically Learn Traffic Classes Using Inside Prefixes: Example, page 60 • Manually Selecting Inside Prefixes for OER Monitoring: Example, page 62 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 2 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 8 • Specifying the Flow Keys for Automatic Learning of Application Traffic Classes, page 36 • Configuring OER to Automatically Learn Traffic Classes Using Protocol, Ports, and DSCP Value: Example, page 60 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 Using OER to Profile the Traffic Classes (continued)

Feature Name	Releases	Feature Configuration Information
OER - Application Aware Routing with Static Application Mapping	12.4(15)T	The OER - Application Aware Routing with Static Application Mapping 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. The following sections provide information about this feature: • Learn List Configuration Mode, page 5 • Static Application Mapping Using OER, page 5 • Application Traffic Class Configuration Using OER, page 8 • Configuring Application Aware Routing to Profile the Traffic Classes in Cisco IOS Release 12.4(15)T and Later Releases, page 9 • Configuring Application Aware Routing to Profile the Traffic Classes in Cisco IOS Release 12.4(15)T and Later Releases: Examples, page 56 The following commands were introduced or modified by this feature: **clear oer master traffic-class**, **count**, **delay (OER)**, **list (OER)**, **match traffic-class access-list**, **match traffic-class application**, **match traffic-class prefix-list**, **show oer master learn list**, **show oer master traffic-class**, **throughput**, **traffic-class access-list**, **traffic-class application, traffic-class prefix-list**.

Table 2 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 • Configuring Application Aware Routing to Profile the Traffic Classes in Cisco IOS Release 12.4(15)T and Later Releases, page 9 • Configuring Application Aware Routing to Profile the Traffic Classes in Cisco IOS Release 12.4(15)T and Later Releases: Examples, page 56 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.