This chapter provides information about configuring Ethernet virtual circuits (EVC) for the Cisco ONS 15454 ML-MR-10 card.

The ML-MR-10 card employs the Cisco IOS Modular QoS command-line interface (CLI), known as the MQC. For more information on general MQC configuration, refer to the following Cisco IOS documents:

- Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.2
- Cisco IOS Quality of Service Solutions Command Reference, Release 12.2

Contents

- 1 Understanding EVC
- 2 Configuring EVC
  - 2.1 Layer 2 Ethernet Services
    - 2.1.1 Restrictions and Usage Guidelines
  - 2.2 Configuring Layer 2
    - 2.2.1 Examples
      - 2.2.1.1 QinQ Point-to-Point EVC
      - 2.2.1.2 VLAN Translation
      - 2.2.1.3 Untagged Service Instance
    - 2.2.2 Default Service Instance
    - 2.2.3 Verification
  - 2.3 EVC QoS Support
    - 2.3.1 Restrictions and Usage Guidelines
  - 2.4 Port Channel QoS
  - 2.5 QoS Classification
    - 2.5.1 Restrictions and Usage Guidelines
  - 2.6 QoS Classifiers Supported on Various Frames on ML-MR-10 Card
  - 2.7 Configuring QoS Traffic Class
  - 2.8 Configuring Policing
    - 2.8.1 Restrictions and Usage Guidelines
      - 2.8.1.1 Table 28-1: Policer actions supported
  - 2.9 Configuring QoS Traffic Policies
    - 2.9.1 Examples
    - 2.9.2 Verification
  - 2.10 Associating a QoS Traffic Policy with an Interface, or Service Instance
    - 2.10.1 Associating a QoS Traffic Policy with an Input Interface
    - 2.10.2 Associating a QoS Traffic Policy with an Output Interface
  - 2.11 Configuring Marking
    - 2.11.1 Restrictions and Usage Guidelines
  - 2.12 Configuring QoS Class-based Marking
    - 2.12.1 Examples
    - 2.12.2 Verification
  - 2.13 Configuring EVC on RPR-IEEE
    - 2.13.1 Restrictions and Usage Guidelines
  - 2.14 Configuring Service Domains
    - 2.14.1 Examples

Understanding EVC

Ethernet virtual connection services (EVCS) uses the concepts of EVCs and service instances to provide Layer 2 switched Ethernet services. An EVC is an end-to-end representation of a single instance of a Layer 2 service being offered by a provider to a customer. It embodies the different parameters on which the service is being offered. A service instance is the instantiation of an EVC on a given port on a given ML-MR-10 card.

Configuring EVC

Establishing EVC on the ML-MR-10 card involves three basic operations:

- Configuring Layer 2 Ethernet services
- (optional) Configuring quality of service (QoS)
- Configuring resilient packet ring (RPR) Interfaces

Layer 2 Ethernet Services

IEEE 802.1Q (QinQ) mapping and service awareness on the ML-MR-10 card provides the following functionality:

- Only point-to-point EVC services are supported in Software Release 8.5
- MAC address learning is not supported in Software R8.5
- QinQ (1-to-2 translation) Layer 2 switching-QinQ adds an outer tag to the received dot1q traffic and then performs Layer 2 switching.
- Local VLAN significance-VLAN tags are significant only to the port.
- VLAN translation is supported

Restrictions and Usage Guidelines

When configuring QinQ Mapping and Service Awareness on ML-MR-10 cards, follow these restrictions and usage guidelines:

- Service Scalability:
  - ♦ Service Instances: 8,000
  - ♦ Bridge-domains: 4,000
- MQC actions supported include:
  - ♦ Bandwidth/Weighted Deficit Round Robbin (WDRR) queuing
  - ♦ One priority queue per policy
  - ♦ Police, set Class of Service (CoS) (marks 802.1p bits)
  - ♦ Police, set QoS-group (egress queue number)
  - ♦ Police, set discard-class (typically on non-edge nodes)
  - ♦ Police, set rpr-ieee service-class {A| B| C}

  Valid for traffic out of RPR interface
  qos-group is ignored for service-class A and B traffic
  qos-group will be considered for service-class C traffic (C0, C1, C2, C3 queues are used for this case)
  - ◊ Police, set discard-class {0| 1| 2} (0 = Green, 1 = Yellow, 2 = Red)

Configuring Layer 2

Use the following commands to configure Layer 2.

Step	Command	Purpose
1	`Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
2	`Router# configure terminal`	Enters global configuration mode.
3	`interface gigabitethernet [port number]`	Specifies the Gigabit Ethernet interface to configure, where: • port-Specifies the location of the interface.
4	`Router(config-if)# [no] service instance id Ethernet [service-name}`	Creates a service instance (an instantiation of an EVC) on an interface and sets the device into the config-if-srv submode.
5	`Router(config-if)#` **`encapsulation dot1q`** `vlan-id`	Defines the matching criteria to be used in order to map ingress dot1q frames on an interface to the appropriate service instance.
6	`Router(config-if)# rewrite egress tag {push {dot1q vlan-id | pop 1 | translate 1-to-1 dot1q vlan-id}`	Specifies the tag manipulation that is to be performed on the frame egress to the service instance.

Examples

QinQ Point-to-Point EVC

In this example, an incoming frame with a dot1q tag of 10 enters GigabitEthernet1 and exits with a dot1q tag of 11. No MAC learning is involved.

```
 !Customer facing port

Router(config)# interface GigabitEthernet1

Router(config-if)# service instance 100 ethernet

Router(config-if-srv)# encapsulation dot1q 10

Router(config-if-srv)# rewrite egress tag pop 1

Router(config-if-srv)# bridge-domain 10

 !RPR IEEE 802.17 Ring facing port

Router(config)# interface rpr-ieee0

Router(config-if)# service instance 101 ethernet

Router(config-if-srv)# encapsulation dot1q 11
```

```
Router(config-if-srv)# rewrite egress tag push 1 dot1q 11

Router(config-if-srv)# bridge-domain 10
```

VLAN Translation

```
 !Customer facing port

Router(config)# interface GigabitEthernet1

Router(config-if)# service instance 100 ethernet

Router(config-if-srv)# encapsulation dot1q 10

Router(config-if-srv)# rewrite egress tag translate 1-to-1 dot1q 10

Router(config-if-srv)# bridge-domain 10

 !RPR IEEE 802.17 Ring facing port

Router(config)# interface rpr-ieee0

Router(config-if)# service instance 101 ethernet

Router(config-if-srv)# encapsulation dot1q 11

Router(config-if-srv)# rewrite egress tag translate 1-to-1 dot1q 11

Router(config-if-srv)# bridge-domain 10
```

Untagged Service Instance

```
 !Customer facing port

Router(config)# interface GigabitEthernet1

Router(config-if)# service instance 11 ethernet

Router(config-if-srv)# encapsulation untagged

Router(config-if-srv)# rewrite egress tag pop 1

Router(config-if-srv)# bridge-domain 11

 !RPR IEEE 802.17 Ring facing port

Router(config)# interface rpr-ieee0

Router(config-if)# service instance 11 ethernet

Router(config-if-srv)# encapsulation dot1q 10

Router(config-if-srv)# rewrite egress tag push dot1q 10
```

```
Router(config-if-srv)# bridge-domain 11
```

Default Service Instance

The Default Service instance can be used to handle the complete traffic on an interface. **Note:** The Default Service instance cannot be configured with any other Service instances. The Default Service instance captures all of the traffic on that interface including tagged, priority tagged, and untagged traffic.

```
 !Customer facing port

Router(config)# interface GigabitEthernet2

Router(config-if)# service instance 12 ethernet

Router(config-if-srv)# encapsulation default

Router(config-if-srv)# rewrite egress tag pop 1

Router(config-if-srv)# bridge-domain 12

 !RPR IEEE 802.17 Ring facing port

Router(config)# interface rpr-ieee0

Router(config-if)# service instance 12 ethernet

Router(config-if-srv)# encapsulation dot1q 12

Router(config-if-srv)# rewrite egress tag push dot1q 12

Router(config-if-srv)# bridge-domain 12
```

Verification

Use the following commands to verify operation.

Command	Purpose
Router# **show ethernet service instance** [**id** *instance-id* **interface** interface-id | interface interface-id] [detail]	Displays information about one or more service instances: If a service instance ID and interface are specified, only data pertaining to that particular service instance is displayed. If only an interface ID is specified, displays data for all service instances s on the given interface.
Router# **show ethernet service interface** [*interface-id*] [**detail**]	Displays information in the Port Data Block (PDB).
Router# **show ethernet service instance** [**id** *instance-id*] [**platform**]	Displays all the Ethernet Flow Points (EFP's) platform information (EFP status and RPR destination) for the card
Router# **show ethernet service interface [platform]**	Displays all that specific interface's EFPs platform info (EFP status and RPR destination)
Router# **show ethernet service instance** [**id** *instance-id*	Displays the specific EFP's platform info (EFP status and RPR destination)

interface *interface-id*] [**platform**]	

EFP status has three possible states.

- UP
  - Both the ingress and egress EFP's physical interfaces IDB state should be UP. (One exception is that when CPP is enabled look at CPP state instead of IDB state)
  - H/W programming (TCAM) is intended for both the ingress and egress EFPs, and configuring both of the EFP's encap, rewrite, bridge domains is a prerequisite to h/w programming
  - When Service Advertisements are enabled for one of the EFPs, then RPR destination should be in the "Resolved" status
- DOWN
  - If any of the above criteria (for declaring the EFP status UP) is not met then the EFP status is declared "DOWN"
- FLAPPING
  - ML-MR-10 supports only point-to-point services and the service/VLAN (VLAN is the S-VLAN or outer tag) must be configured on exactly two RPR stations. In case of a network mis-configuration in which more than two RPR stations are configured to have the same service/S-VLAN, the Service Advertisement scheme will advertise two or more destinations for the same service. In this scenario the EVC platform can detect, and will declare, EFP status as "FLAPPING".

Note: An error message automatically displays on the console when there is a "FLAPPING" service. This is useful to determine which service is FLAPPING. When the EFP is in "FLAPPING" status it is service affecting because for a few seconds traffic goes to one destination and then another destination, and follows a circular path as multiple destinations keep advertising one after the other - the same service/VLAN

The RPR destination field is valid only for the EFPs that are configured on RPR interfaces. It provides three possible types of output:

- Unresolved
  - When the Service Advertisement can not resolve the RPR destination for the point-to-point service because it is mis-configured, or due to operational errors (like Interface down) then the RPR destination are displayed as "Unresolved"
- mac-address> (learnt)
  - When the Service Advertisement scheme is able to resolve the RPR destination, the MAC address, together with the keyword "learnt," are displayed.
- mac-address> (static)
  - When the RPR destination is configured statically (using rpr-destination under service instance mode) the MAC address together with the keyword "static," are displayed.

Sample Output

```
!show ethernet service instance plat
Router#show ethernet service instance platform
NOTE: EFP status UP/DOWN is determined based on both ingress and egress
interface states and RPR destination resolving status. EFP status FLAPPING
means more than one RPR station is advertising this specific P2P service
and need to check the network level config.
(*) RPR-destination field is valid only for EFPs configured on RPR interfaces
EFP-ID     Intf  EFP-Status RPR-Destination
1          Gi0   DOWN       Not applicable
1          Gi8   DOWN       Not applicable
30         Gi8   DOWN       Not applicable
```

```
30          RP0   UP          aabb.bbbb.cccc (static)
```

EVC QoS Support

Refer to Configuring Quality of Service for information on QoS and for additional details on configuring the ML-Series cards.

Restrictions and Usage Guidelines

When configuring QoS with EVCS on the ML-MR card, follow these restrictions and usage guidelines:

- Service instances use MQC.
- QoS supports 4, 000 service instances.
- For ingress and egress QoS, only flat policy maps are supported.
- As service instance configurations change, Cisco recommends that you remove and re-apply the policy because the policy map configuration may no longer be valid.
- Ethernet Flow Points (EFP) are supported on port channels

EVC QoS supports:

- EVC QoS, classification is based on the following filters, which can be combined:
  - Inner VLAN tag
  - Outer VLAN tag
  - CoS
- Egress Qos Policy supports the bandwidth and priority actions
  - Bandwidth command - assigns the Queue in WDRR mode, and the bandwidth value can either be absolute value or a percentage value
  - Priority command - assigns the Queue in strict priority mode, no bandwidth value needs to be associated with this queue
  - For RPR interfaces, only bandwidth in percentage values is supported for the bandwidth command
  - For port-channel interfaces it is recommended that you use percentage values for the bandwidth command, as the total bandwidth of a port-channel varies, based on member availability
- EVC QoS, actions supported:
  - Supports up to 4 queues per port scheduled with a Weighted Deficit Round Robin (WDRR) scheduler
  - One egress strict priority queue per port
  - There are two modes of operation for the egress queues: all four queues operate in WDRR mode, or one queue is in strict priority mode and the other three are in WDRR mode
  - Mappings between the cos index values and the egress queue are allowed and achieved using the qos-group value
  - Re-marking of the 802.1p VLAN priority bits in a frame is supported
- Policing:
  - A 2-rate 3-color policer is supported
  - The two rates are cir (committed information rate) and pir (peak information rate)
  - The two burst sizes corresponding to the rates are cbs (committed burst size) and pbs (peak burst size)
  - The 3 colors correspond to the possible outcomes of the policer (conform, exceed and violate)
  - Conform actions are transmit and set cos-transmit
  - Exceed actions are drop and set cos-transmit
  - Violate actions are drop and set cos-transmit

- MOQ (Modular QOS CLI)
  - Service policy is configurable on a physical Ethernet interface in the ingress direction
  - Service policy is configurable on a physical Ethernet interface in the egress direction
  - Service policy is configurable on an etherchannel/ port channel bundle interface in the ingress direction
  - Service policy is configurable on an etherchannel/ port channel bundle interface in the egress direction
  - If a physical interface is a member of an etherchannel/ link aggregation bundle it will not allow configuration of any MQC service policy (service policies mst be configured on the logical etherchannel/ link aggregation bundle interface only)
  - MQC service policy is configurable on an Ethernet Flow Point (EFP) in the ingress direction
  - Match cos 0-7 is supported on ingress service policies attached to physical Ethernet interfaces, and to EFPs, as long as the type of the EFP is not untagged (supports packet classification based on the 802.1p priority bits in the outermost 802.1Q VLAN header in the packet)
  - Match ip dscp 0-63 is supported on physical interfaces in the ingress direction
  - Match ip precedence 0-7 is supported on physical interfaces in the ingress direction
  - Match ip dscp 0-63 is supported on EFPs in the ingress direction, as long as the type of the EFP is untagged
  - Match ip precedence 0-7 is supported on EFPs in the ingress direction as long as the type of the EFP is untagged
  - Set cos 0-7 is supported as a possible action in an ingress service policy (interpreted to mean setting of the 802.1p bits in the outermost 802.1Q header of the packet (if there is one) when the packet goes out the egress port)
  - pir or pbs are set equal to cir and cbs
  - All actions in an egress service policy configured on an etherchannel bundle interface are applied equally across all members of the bundle (and not in an aggregate as on the ingress side). For example, it is not correct to calculate the egress bandwidth that can be reserved on an etherchannel bundle to be equal to the sum of the bandwidths of the individual members interfaces, instead, it is the minimum of the bandwidths of any of the members

Port Channel QoS

QoS is supported on Port channel Interfaces, and Service policies are applied on the port channel interface, instead of a member interface. At the Ingress, the QoS actions are applied on the aggregate interface. At the Egress, the bandwidth guaranteed on a Port channel interface is limited to one member interface's bandwidth. Policing can be configured up to the maximum port channel bandwidth.

QoS Classification

Use the QoS classification features to select your network traffic and categorize it into classes for further QoS processing based on matching certain criteria. The default class, named "class-default," is the class to which traffic is directed for any traffic that does not match any of the selection criteria in the configured class maps. **Note:** When class-default is applied on a physical interface, any traffic on the interface, regardless of the EVC, matches this class. When "class-default" is applied on an EFP, any traffic on this EFP matches the class.

Restrictions and Usage Guidelines

When configuring traffic classes on an ML-MR-10 card, follow these restrictions and usage guidelines:

- You can define up to four unique class maps per output Service Policy.
- You can define up to 65 class maps on ingress.

QoS Classifiers Supported on Various Frames on ML-MR-10 Card

Following are QoS classifiers supported on various frames on the ML-MR-10 card:

Supported Ethernet Frame Types on ML-MR-10	Frame Type Seen by Classifier	Fields Used for Classification
Untagged IP frame	Untagged IP	IP Precedence/DSCP depending on the class map configured on the interface
Untagged SNAP[1]	Untagged SNAP	IP Precedence/DSCP depending on the class map configured on the interface
Tagged IP frame	Tagged IP	IP Precedence/DSCP or VLAN COS value depending on the class map configured on the interface
Tagged SNAP frame	Tagged SNAP	IP Precedence/DSCP or VLAN COS value depending on the class map configured on the interface
Q-in-Q	802.1Q	Outer Q-in-Q VLAN COS value based on the configuration on the interface

1. SNAP = Subnetwork Access Protocol.

Configuring QoS Traffic Class

To create a user-defined QoS traffic class, use the following commands, beginning in global configuration mode:

Step	Command	Purpose
1	`Router(config)# class-map [match-all] class-name`	Creates a traffic class, where: • **match-all**-(Optional) Specifies that all match criteria in the class map must be matched, using a logical AND of all matching statements defined under the class. This is the default. • class-name-Specifies the user-defined name of the class. **Note:** You can define one match criteria per unique class map.
2	`Router(config-cmap)# match type`	Specifies the matching criterion to be applied to the traffic, where type represents one of the forms of the **match** command supported by the ML-MR-10 card. **Note:** In ingress service policy match cos, match ip dscp, match ip precedence, and match any are allow. **Note:** On egress interface policy, match qos-group, and match any are allowed.

Configuring Policing

This section describes information for configuring QoS traffic policing policies.

Restrictions and Usage Guidelines

The ML-MR-10 supports different forms of policing using the **police** command.

When configuring ingress policing on interfaces, and VLANs, follow these restrictions and usage guidelines:

- Policing on physical interfaces is supported
- Policing on service instances is supported
- Policing supports three actions:
  - Transmit
  - Set-cos transmit
  - Drop
- Set-dscp-transmit is not supported

See Table 28-1 for Policer action details.

Table 28-1: Policer actions supported

	Single rate two color Policer		**Single/Two rate three color Policer**		
Action/Condition	Conform	Exceed/Violate	Conform	Exceed	Violate
Transmit	Supported	Not supported	Supported	Not supported	Not supported
set-cos-transmit	Supported (Alternative - set-cos action can be used)	Supported	Supported (Alternative - set-cos action can be used)	Supported	Supported
Drop	Not supported	Supported	Not supported	Not supported	Supported

Configuring QoS Traffic Policies

To create QoS traffic policies with policing, use the following commands beginning in global configuration mode:

Note: When creating QoS traffic policies as shown below, you can perform only one of the commands shown in Step 3 and Step 4 for each policy. You can perform Step 3 or Step 4; do not attempt to perform Step 3 and Step 4 in the same policy.

Step	Command	Purpose
1	`Router(config)# policy-map policy-map-name`	Creates or modifies a traffic policy and enters policy map configuration mode, where: • policy-map-name-Specifies the name of the traffic policy to configure. Names can be a maximum of 40 alphanumeric characters.
2	`Router (config-pmap)# class {class-name | class-default}`	Specifies the name of the traffic class to which this policy applies and enters

		policy-map class configuration mode, where: • class-name-Specifies that the policy applies to a user-defined class name previously configured. • **class-default**-Specifies that the policy applies to the default traffic class.
3	`Router(config-pmap-c)# police bps [burst-normal] [burst-max] conform-action action exceed-action action violate-action action`	Specifies a maximum bandwidth usage by a traffic class through the use of a token bucket algorithm, where: • bps-Specifies the average rate in bits per second. Valid values are 1000000 to 4000000000. • burst-normal-(Optional) Specifies the normal burst size in bytes. Valid values are 16000 to 128000000. The default normal burst size is 16000 bytes. • burst-max-(Optional) Specifies the excess burst size in bytes. Valid values are 16000 to 128000000. • action-Specifies the policing command (as shown in Table: Policer actions supported) for the action to be applied to the corresponding conforming, exceeding, or violating traffic. • 200 ms of burst is recommend when configuring rates
Or		
4	`Router(config-pmap-c)# police {cir cir} [bc conform-burst] {pir pir} [be peak-burst] [conform-action action [exceed-action action [violate-action action]]]`	Configures traffic policing using two rates, the committed information rate (CIR) and the peak information rate (PIR), where: • **cir** cir-Specifies the CIR at which the first token bucket is updated as a value in bits per second. The value is a number from 8000 to 200000000. • **bc** conform-burst-(Optional) Specifies the conform burst (bc) size in bytes used by the first token bucket for policing. The value is a number from 1000 to 31,250,000. • **pir** pir-Specifies the PIR at which the second token bucket is updated as a value in bits per second. The value is a number from 8000 to 200000000. • **be** peak-burst-(Optional) Specifies the peak burst (be) size in bytes used by the second token bucket for policing. The size varies according to

		the interface and platform in use. • action-(Optional) Specifies the policing command (as shown in Table 28-1) for the action to be applied to the corresponding conforming, exceeding, or violating traffic.

Examples

Input service policy can be applied to a Physical Interface, or to the Service Instance.

This example shows Classmap configuration

```
Router(config)# class-map match-all voip_class

Router(config-cmap)# match cos 5

 !

Single Rate Two color Policer

Router(config)# policy-map example_policy

Router(config-pmap)# class voip_class

Router(config-pmap-c)# police 1000000 25000 conform-action transmit
exceed-action drop

 !

Two rate three color marker example

Router(config)# policy-map remark_policy

Router(config-pmap)# class voip_class

Router(config-pmap-c)# police 50000000 1250000 2500000 pir 100000000
conform-action transmit exceed-action set-cos-transmit 3 violate-action
drop
```

Verification

Use the following commands to verify policing:

Command	Purpose
`Router#` **`show policy-map`**	Displays all configured policy maps.
`Router#` **`show policy-map`** *`policy-map-name`*	Displays the user-specified policy map.
`Router#` **`show policy-map interface`**	Displays statistics and configurations of all input and output policies that are attached to an interface.

This example shows an Ingress Service Police with two color policer action specified.

```
Router# show policy-map interface 7
```

```
service-policy input: in

class-map: i1(match-all)

33239921 packets, 3058072732 bytes

5 minute rate 70861000 bps, drop rate 776000 bps

match: cos 1

police:

2000000 bps, 50000 limit

conformed 120206 packets, 11895232 bytes; actions: transmit

exceeded 33110625 packets, 3046177500 bytes; action: drop

class-map: class-default (match-all)

0 packets, 0 bytes

5 minute offered rate 0 bps, drop rate 0 bps

match: any

20 packets, 0 bytes

5 minute 0 bps
```

This example displays an Egress Service Policy.

```
Router# show policy-map interface 9

GigabitEthernet9

Service-policy output: 1

Counters last updated 00:00:00 ago

class-map: 1 (match-all)

3 packets, 1014 bytes

5 minute offered rate 0 bps, drop rate 0 bps

match: qos-group 2

class-map: 2(match-all)

0 packets, 0 bytes

5 minute offered rate 0 bps, drop rate 0 bps
```

```
match: qos-group 1

class-map: class-default (match-all)

129296 packets, 12925268 bytes

5 minute offered rate 270000 bps, drop rate 0 bps

match: any

3 packets, 990 bytes

5 minute 0 bps
```

Associating a QoS Traffic Policy with an Interface, or Service Instance

Before a traffic policy can be enabled for a class of traffic, it must be configured on an interface. A traffic policy also can be associated with Ethernet service instances

Traffic policies can be applied for traffic coming into an interface (input) and for traffic leaving that interface (output).

Traffic policies can not be applied for output on service instance.

Associating a QoS Traffic Policy with an Input Interface

When you associate a traffic policy with an input interface, the policy is applied to traffic coming into that interface. To attach a traffic policy for an input interface, use the following command beginning in interface configuration mode:

Command	Purpose
`Router(config-if)# service-policy input policy-map-name`	Attaches a traffic policy to the input direction of an interface, where: • policy-map-name-Specifies the name of the traffic policy to configure.

Associating a QoS Traffic Policy with an Output Interface

When you associate a traffic policy with an output interface, the policy is applied to traffic leaving that interface. To attach a traffic policy to an output interface, use the following command beginning in interface configuration mode:

Command	Purpose
`Router(config-if)# service-policy output policy-map-name`	Attaches a traffic policy to the output direction of an interface, where: • policy-map-name-Specifies the name of the traffic policy to configure.

Configuring Marking

After you have created your traffic classes, you can configure traffic policies to configure marking features to apply certain actions to the selected traffic in those classes.

In most cases, the purpose of a packet mark is identification. After a packet is marked, downstream devices identify traffic based on the marking and categorize the traffic according to network needs. This categorization occurs when the **match** commands in the traffic class are configured to identify the packets by the mark (for example, **match ip precedence**, **match ip dscp**, **match cos**, and so on). The traffic policy using this traffic class can then set the appropriate QoS features for the marked traffic.

Restrictions and Usage Guidelines

When configuring class-based marking on an ML-MR-10 cards, follow these restrictions and usage guidelines:

- Marking is supported two ways:
  - policing
  - class-based marking

Configuring QoS Class-based Marking

To configure a QoS traffic policy with class-based marking, use the following commands beginning in global configuration mode:

Step	Command	Purpose
1	`Router(config)# policy-map policy-map-name`	Creates or modifies a traffic policy and enters policy map configuration mode, where: • policy-map-name-Specifies the name of the traffic policy to configure. Names can be a maximum of 40 alphanumeric characters.
2	`Router (config-pmap)# class {class-name | class-default}`	Specifies the name of the traffic class to which this policy applies and enters policy-map class configuration mode, where: • class-name-Specifies that the policy applies to a user-defined class name previously configured. • **class-default**-Specifies that the policy applies to the default traffic class.
3	`Router(config-pmap-c)# set cos [0-7]`	S802.1p bits can be marked.

Examples

This example shows the creation of a service policy called policy1. This service policy is associated to a previously defined classification policy through the use of the class command. This example assumes that a classification policy called class1 was previously configured.

```
Router(config)# policy-map policy1

Router(config-pmap)# class class1
```

```
Router(config-pmap-c)# set cos 3
```

Verification

Use the following commands to verify marking:

Command	Purpose
`Router# show policy-map`	Displays all configured policy maps.
`Router# show policy-map policy-map-name`	Displays the user-specified policy map.
`Router# show policy-map interface`	Displays statistics and configurations of all input and output policies that are attached to an interface.

For more detailed information about configuring class-based marking features, refer to the *Class-Based Marking* document located at the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121t/121t5/cbpmark2.htm

Configuring EVC on RPR-IEEE

Refer to Configuring IEEE 802.17b Resilient Packet Ring, for information on rpr-ieee. Two commands are available; configure service advertisements, and configure static RPR destinations. Begin in global command mode and use the following commands.

Restrictions and Usage Guidelines

When configuring RPR features on an ML-MR-10 card, follow these restrictions and usage guidelines:

- All packets use bandwidth only between the source and destination nodes on the ring associated with the EVC
- Flooding of broadcast packets or unknown destination packets is not supported. The RPR connections are point to point only for each frame associated with a given EVC
- An EVC will have 2 EFPs on the RPR ring, both EFPs share a unique VLAN tag
- The EFP VLAN tag is added to the frame, (along with the source and destination RPR node MAC addresses), when the frame is sent to the RPR ring. The EFP VLAN is removed from the frame, (along with the source and destination RPR node MAC addresses), when the frame is received from the RPR ring
- Flooding of broadcast packets or unknown destination packets is not supported on point to point EVCs.
- RPR connections are point to point only for each frame associated with a given point to point EVC
- Attribute discovery frames (ATD) are used to advertise remote EFPs for each EVC on the RPR ring
- Source MAC address learning of destination RPR nodes is not supported - ATD frames are used to determine the EFP mapping to the remote RPR node
- EVC Frames associated with an EFP are dropped and not sent to the RPR ring if there has been no ATD frame received for the remote EPF associated with the EVC
- Traffic can be put in bandwidth queues based on QOS classification via MQC. This along with assigning an MQC class to an 802.17 Class of Service provides unequal local fairness within an 802.17 Class

Step	Command	Purpose
1	`Router(config)# interface rpr-ieee 0`	Activates interface configuration mode to configure the RPR-IEEE

		interface.
2	`Router(config-if)#service instance 50 ethernet`	Specifies the service instance Ethernet Flow Point (EFP)
3	`Router(config-if-srv)# bridge-domain 1`	Specifies the EFP Bridge Domain
4	`Router(config-if-srv)# encapsulation`	Configures Ethernet frame match criteria
5	`Router(config-if-srv)#rewrite`	Configures Egress rewrite criteria
6	`Router(config-if-srv)#rpr-destination service-advertisement`	Default configuration (Each service learns the RPR destination using the service advertisements)
7	`Router(config-if-srv)#service-policy policy_name`	Attaches a service policy-map to the EFP
8	`Router(config-if-srv)#rpr-destination static xxxx.xxxx.xxxxx`	48-bit hardware address of the RPR destination

Configuring Service Domains

The following example shows a service domain configuration.

Note: All of the existing services, except services with static RPR destinations configured, will be interrupted and re-assigned to the new domain-id

Examples

```
 !

Router(config)interface RPR-IEEE0

Router(config-if) no ip address

Router(config-if) no ip route-cache

Router(config-if) no rpr-ieee sas

Router(config-if) rpr-ieee vlan-service-domain 10

Router(config-if-srv) service instance 10

Router(config-if-srv) ethernet encapsulation dot1q 10

Router(config-if-srv) bridge-domain 10

 !
```